DEFINIÇÃO

Componentes do modelo relacional. Formas normais e normalização de dados.
Mapeamento conceitual-lógico: entidades, relacionamentos, atributos e especialização/generalização. Diretrizes para implementação do modelo no SGBD (Sistema de Gerenciamento de Banco de Dados).

PROPÓSITO

Conhecer os elementos do modelo relacional e as formas normais é essencial para aprender sobre as regras utilizadas no mapeamento conceitual-lógico. É importante compreender os aspectos físicos que influenciam a implementação do modelo no SGBD, pois são atividades da rotina dos profissionais de banco de dados.

PREPARAÇÃO

É recomendável que você reproduza os exemplos práticos usando uma ferramenta para modelagem de dados. Certifique-se de ter baixado para seu computador a ferramenta livre BrModelo.

OBJETIVOS

MÓDULO 1

Identificar os elementos do modelo relacional

MÓDULO 2

Diferenciar formas normais

MÓDULO 3

Aplicar o mapeamento conceitual-lógico

MÓDULO 4

Identificar aspectos físicos para implementação do modelo no SGBD

INTRODUÇÃO

Ao longo deste tema, vamos conhecer os principais conceitos e componentes do modelo relacional. Aprenderemos que o modelo relacional representa o banco de dados sob o formato de tabelas, estando presente em diversos sistemas gerenciadores de banco de dados (SGBD), tais como MySQL, Oracle, PostgreSQL e SQL Server.

Ainda, vamos estudar o processo de Normalização como forma de avaliar a qualidade de um projeto de banco de dados relacional. Em seguida, aprenderemos regras que deverão ser aplicadas para obtermos um modelo lógico a partir de um modelo conceitual.

Finalmente, investigaremos alguns aspectos físicos que devem ser levados em consideração na implementação do modelo no SGBD.

MÓDULO 1

- Identificar os elementos do modelo relacional

MODELO RELACIONAL

Relação é um termo usado na literatura formal da área de banco de dados. No contexto comercial, usa-se informalmente o termo tabela.

O MODELO RELACIONAL REPRESENTA O BANCO DE DADOS COMO UMA COLEÇÃO DE RELAÇÕES

(ELMASRI; NAVATHE, 2019).

COMPONENTES DE UMA TABELA

Uma tabela corresponde a um conjunto não ordenado de linhas, que, na terminologia acadêmica, são conhecidas por *tuplas* .

As linhas de uma tabela são divididas em campos ou colunas, que, na academia, são chamados de atributos. Os campos são nomeados com objetivo de facilitar a interpretação dos dados armazenados.

Observe a tabela ALUNO a seguir:

CODIGOALUNO	NOME	NOMEMAE	CPF	DTNASCIMENTO
1	Aline Gonçalves Campos		09320900022	13/02/1980
2	Pablo Gonçalves Campos	Maria Augusta Gonçalves	08760900022	13/12/1984
3	Bruno da Silva	Yvone Silva	99920900099	15/02/1990
4	Viviane da Silva	Yvone Silva	00209000922	15/07/1994
5	Lucas Pontes Silva	Daniele Pontes Maciel	11109000933	15/07/1994

Fonte: Autor

Figura: Componentes de uma tabela de banco de dados.

NOME DE TABELAS

Em um banco de dados relacional, toda tabela deve possuir um nome único. Além disso, ao longo do nosso estudo, vamos perceber que a maioria das tabelas de um banco de dados representa entidades de um diagrama de entidade e relacionamento (DER).

ATENÇÃO

É importante que, na medida do possível, o nome da tabela represente com clareza o objeto modelado. Por exemplo, ao lermos o nome ALUNO, criamos a expectativa natural de que a tabela em questão armazene informações sobre alunos.

COLUNAS DE TABELAS

A primeira linha da tabela de exemplo contém os seguintes campos ou cabeçalhos de coluna: CODIGOALUNO, NOME, NOMEMAE, CPF e DTNASCIMENTO. Além disso, o nome de coluna deve ser único em cada tabela.

Com isso, percebe-se que o nome de coluna ajuda a entender o papel ou a finalidade dela. Por exemplo, podemos concluir que DTNASCIMENTO identifica a data de nascimento do aluno.

Outro ponto é que as colunas de uma tabela são monovaloradas, ou seja, é permitido manter no máximo um item de informação por vez. Por exemplo, é possível existir até uma ocorrência de data de nascimento na coluna DTNASCIMENTO.

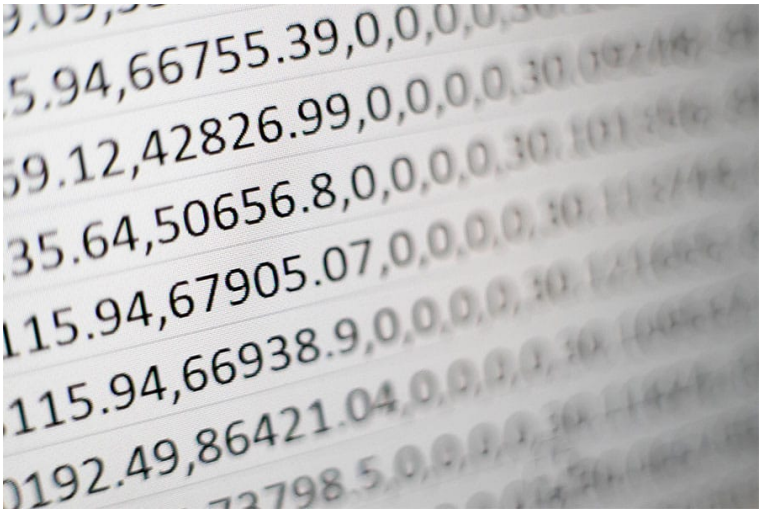
As colunas de uma tabela possuem valores atômicos, ou seja, não admitem colunas compostas de outras. Por exemplo, não é possível subdividir CODIGODOALUNO em outros campos.

Ao implementar uma tabela em um banco de dados, é necessário definir um tipo de dado para cada coluna. Os mais comuns são: caractere, numérico, data e booleano.

Alguns SGBDs (Sistemas de Gerência de Banco de Dados) permitem a definição do tipo de dados feita pelo usuário. Em uma linguagem mais técnica, o conjunto de valores que uma coluna pode assumir é denominado **domínio da coluna ou domínio do campo**.

Quando criarmos uma tabela, devemos definir se o valor da coluna é opcional ou obrigatório, em que especificar que uma coluna é opcional significa que os valores admitem vazio (NULL) ou nulo. Na tabela ALUNO, a coluna NOMEMAE da linha correspondente à aluna de código 1 está vazia.

LINHAS DE TABELAS



Fonte: Mika Baumeister/Unsplash

As linhas da tabela, da segunda em diante, representam um item de informação cadastrado no banco de dados. Dizemos então que um item de informação corresponde a uma unidade básica que servirá para armazenamento e recuperação de dados. Isto é, se uma tabela de cadastro de alunos contém 10.000 linhas ou registros, podemos dizer que ela armazena dados de 10.000 alunos.

As linhas de uma tabela permitem o armazenamento de dados sempre de acordo com a semântica ou o significado do objeto. No caso em tela, a tabela armazena informações de ALUNOS, portanto queremos dizer que essa mesma tabela não deve ser utilizada para armazenar outros tipos de objetos, como disciplinas ou docentes.

CHAVE PRIMÁRIA

Em uma tabela, um SGBD precisa diferenciar uma linha das demais, isso é feito a partir da definição de uma restrição de integridade. Na prática, escolheremos uma ou mais coluna(s) para que seu(s) valores se torne(m) únicos no banco de dados.

📢 ATENÇÃO

Quando escolhemos uma coluna para ser chave primária, dizemos que estamos diante de uma chave simples. Se escolhermos mais de uma coluna, a chave é dita composta.

VAMOS ESTUDAR UM EXEMPLO DE CHAVE PRIMÁRIA SIMPLES?

Para diferenciar um estudante dos demais na tabela ALUNO, podemos estabelecer a restrição de chave primária associada à coluna CODIGOALUNO. Ao fazermos isso, na prática, estamos delegando ao SGBD a responsabilidade de gerenciar essa restrição durante todo o ciclo de vida do banco de dados. Na tabela a seguir, deixamos em destaque a coluna CODIGOALUNO:

CODIGOALUNO	NOME	NOMEMAE	CPF	DTNASCIMENTO
1	Aline Gonçalves Campos		09320900022	13/02/1980
2	Pablo Gonçalves Campos	Maria Augusta Gonçalves	08760900022	13/12/1984
3	Bruno da Silva	Yvone Silva	99920900099	15/02/1990

4	Viviane da Silva	Yvone Silva	00209000922	15/07/1994
5	Lucas Pontes Silva	Daniele Pontes Maciel	11109000933	15/07/1994

Figura: Tabela ALUNO com destaque à coluna CODIGOALUNO, escolhida como **chave primária simples**.

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

Assim, podemos dizer que toda chave primária tem as seguintes propriedades:

UNICIDADE

o valor da chave primária não permite repetição;

MONOVALORADO

toda linha da tabela possui no máximo um valor de chave primária;

OBRIGATÓRIO

toda linha da tabela necessariamente tem que ter um valor para a coluna que é chave primária. Em outras palavras, nenhum valor de chave primária deve ser vazio. Esta propriedade é conhecida por **restrição de integridade de entidade**.

Ao longo do nosso estudo, iremos aprender que alguns SGBDs permitem associar uma propriedade especial a um campo, denominada **autoincremento**. Por meio dessa propriedade, o SGBD incrementa automaticamente o valor de uma coluna quando um registro é adicionado à tabela. Trata-se de um mecanismo útil para gerar um valor único para cada registro.

AGORA, VAMOS ESTUDAR UM EXEMPLO CONTENDO CHAVE PRIMÁRIA COMPOSTA.

Considere a tabela a seguir, que representa dados de dependentes de funcionários:

DEPENDENTE

CODIGOFUNCIONARIO	NRDEPENDENTE	NOME	DTNASCIMENTO
1	**1**	Andrey Campos	13/07/2019
1	**2**	Manoel Oliveira	13/12/2018
2	**1**	João Silva	15/02/2017
2	**2**	José Maciel	15/07/2016

Figura: Tabela DEPENDENTE com destaque às colunas CODIGOFUNCIONARIO, NRDEPENDENTE escolhida como chave primária composta.

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

A tabela possui uma chave primária composta pelo par de colunas CODIGOFUNCIONARIO, NRDEPENDENTE. Devemos notar que nenhuma das colunas que compõem a chave é suficiente para, isoladamente, diferenciar uma linha das demais, visto que:

Um CODIGOFUNCIONARIO pode aparecer em diferentes linhas da tabela;

Um NRDEPENDENTE pode aparecer em diferentes linhas da tabela.

CHAVE MÍNIMA

Uma chave primária deve ser mínima. Com isso, queremos dizer que todas as colunas que a formam devem ser necessárias e suficientes para diferenciar uma linha das demais na tabela. Outro ponto importante é que uma chave mínima não diz respeito ao quantitativo de colunas que a forma.

A chave primária simples (coluna CODIGOALUNO) da tabela ALUNO é mínima. Cada valor de CODIGOALUNO é suficiente para diferenciar um aluno dos demais. Perceba que, se decidíssemos definir o par de colunas CODIGOALUNO, CPF como chave primária composta para a tabela ALUNO, a chave não seria mínima, visto que CODIGOALUNO por si só é suficiente para diferenciar um estudante dos outros.

A chave primária composta (CODIGOFUNCIONARIO, NRDEPENDENTE) da tabela DEPENDENTE é mínima. Somente a coluna CODIGOFUNCIONARIO não diferencia um dependente dos demais, pois um funcionário pode ter diversos dependentes. De modo semelhante, somente a coluna NRDEPENDENTE não diferencia um dependente dos outros, dado que o valor dela aparece em mais de uma linha da tabela DEPENDENTE.

CHAVE CANDIDATA

Ao projetarmos uma tabela, pode ser que mais de uma coluna sirva para diferenciar uma linha das demais. Por exemplo, na tabela ALUNO, tanto CODIGOALUNO quanto CPF poderiam ser utilizados como chave primária. Logo, podemos dizer que CODIGOALUNO e CPF são chaves candidatas.

CHAVE ALTERNATIVA

A partir do momento em que escolhemos CODIGOALUNO para ser a chave primária da tabela ALUNO, passamos a considerar CPF como uma chave alternativa.

Alguns desenvolvedores preferem escolher para chave primária uma coluna *artificial* , ou seja, que não tenha dependência das colunas criadas, com objetivo de manter alguma informação referente ao negócio sendo modelado.

CHAVE ESTRANGEIRA

Um banco de dados relacional é composto por um conjunto de tabelas. Na maioria dos casos, existe algum tipo de relacionamento entre elas. O relacionamento entre tabelas é um dos conceitos fundamentais em projeto de banco de dados relacionais.

Modelo relacional

Decorre do termo relação, função matemática que conhecemos popularmente como tabela.

Relacionamento

Relacionamento entre tabelas.

Em geral, uma empresa possui informações sobre funcionários e seus dependentes. Vamos observar então a figura a seguir:

FUNCIONARIO

CODIGOFUNCIONARIO	NOME	SEXO	CPF	DTNASCIMENTO
1	José Maciel	M	23456239999	13/09/1982
2	Pedro Antônio	M	98789345552	13/11/1986
3	Debora Silva	F	12332149048	15/02/1998
4	Amanda de Miranda	F	60989893223	15/05/1997

DEPENDENTE

CODIGOFUNCIONARIO	NRDEPENDENTE	NOME	DTNASCIMENTO
1	1	Andrey Campos	13/07/2019
1	2	Manoel Oliveira	13/12/2018
2	1	João Silva	15/02/2017
2	2	José Maciel	15/07/2016

Fonte: Autor

Figura: Relacionamento entre FUNCIONARIO e DEPENDENTE.

A figura permite as seguintes interpretações:

Todo dependente está associado a um funcionário. Por exemplo, o valor (**1**) de CODIGOFUNCIONARIO na tabela DEPENDENTE permite identificar o funcionário responsável, neste caso **José Maciel**;

Um funcionário pode ter diversos dependentes. Por exemplo, o valor (**2**) de CODIGOFUNCIONARIO nas duas últimas linhas da tabela DEPENDENTE permite concluir que o funcionário **Pedro Antônio** possui dois dependentes;

Um funcionário pode não ter dependentes. Por exemplo, o valor (**3**) de CODIGOFUNCIONARIO na tabela FUNCIONARIO não aparece na tabela DEPENDENTE.

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

UMA CHAVE ESTRANGEIRA É UMA COLUNA OU UMA COMBINAÇÃO DE COLUNAS, CUJOS VALORES APARECEM NECESSARIAMENTE NA CHAVE PRIMÁRIA DE UMA TABELA.

(HEUSER, 2009).

ATENÇÃO

No exemplo, CODIGOFUNCIONARIO da tabela DEPENDENTE é chave estrangeira, pois todo valor dessa coluna necessariamente aparece como valor da coluna CODIGOFUNCIONARIO, chave primária de FUNCIONARIO. Como consequência, podemos concluir que todo dependente está vinculado a um funcionário.

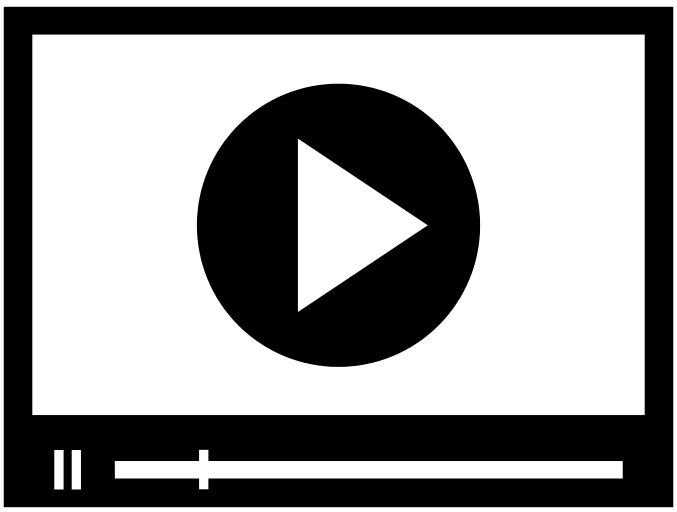
Deve-se notar que, mesmo que a coluna CODIGOFUNCIONARIO de DEPENDENTE tenha o mesmo nome da coluna que é chave primária em FUNCIONARIO, é possível nomeá-la de forma distinta. No entanto, usar o mesmo nome facilita na identificação das colunas relacionadas.

RESTRIÇÕES IMPOSTAS PELA CHAVE ESTRANGEIRA

Percebemos que a chave estrangeira serve para implementar relacionamentos entre tabelas. Para que o banco de dados permaneça íntegro, algumas restrições devem ser controladas e obedecidas pelo SGBD:

Inclusão de linha na tabela que possui chave estrangeira: o sistema deve garantir que o valor da chave estrangeira exista como valor da coluna da chave primária referenciada. Em nosso exemplo, ao incluir um dependente, a coluna CODIGOFUNCIONARIO só pode assumir um dos valores (1,2,3,4).
Alteração de valor da chave estrangeira: o sistema deve garantir que o novo valor da chave estrangeira exista como valor de coluna da chave primária referenciada. Em nosso exemplo, ao alterar um responsável de algum dependente, a coluna CODIGOFUNCIONARIO só pode assumir um dos valores (1,2,3,4).
Exclusão de linha em tabela que contém chave primária referenciada pela chave estrangeira: o sistema deve garantir que todo valor de chave estrangeira sempre faça referência para o valor de alguma chave primária. Em nosso exemplo, os funcionários que têm código 1 ou 2 não devem ser excluídos, pois possuem vínculo com a tabela de dependentes.
Alteração de valor da chave primária referenciada pela chave estrangeira: o sistema deve garantir que o novo valor da chave primária da tabela principal seja replicado nas respectivas dependências. Em nosso exemplo, se alterarmos o valor de CODIGOFUNCIONARIO (tabela FUNCIONARIO) de 2 para 5, o sistema deve garantir a propagação da atualização nas duas últimas linhas de DEPENDENTE.
Os exemplos estudados representam o que conhecemos por integridade referencial, ou seja, os valores de chave estrangeira devem aparecer na chave primária da tabela referenciada.

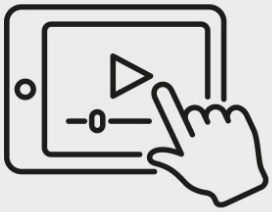
Atenção! Para visualização completa da tabela utilize a rolagem horizontal



RESTRIÇÃO DE INTEGRIDADE REFERENCIAL

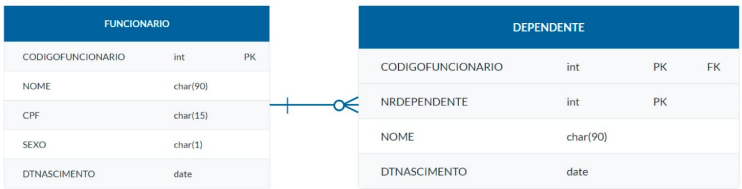
Veja mais exemplos no vídeo a seguir:

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



ESQUEMA DIAGRAMÁTICO DE BANCO DE DADOS RELACIONAL

Diversas ferramentas de modelagem permitem o uso de alguma notação gráfica para representar um banco de dados relacional. Na figura a seguir, temos um diagrama que foi construído a partir de uma ferramenta comercial.



Fonte: Autor

Figura: Esquema diagramático “pé de galinha” envolvendo as tabelas FUNCIONARIO e DEPENDENTE.

A notação utilizada é conhecida por pé de galinha (Do inglês: Crow’s Foot) . Nesse esquema diagramático:

Cada tabela é representada por um retângulo com duas divisões;
Na primeira subdivisão, adicionamos o nome da tabela;
Na segunda subdivisão, aparecem as colunas da tabela, com as informações sobre o nome, o tipo de dados, além de símbolos representativos de chave primária PK (do Inglês, *Primary Key*) e chave estrangeira FK (do Inglês, *Foreign Key*);
O símbolo “colado” na tabela DEPENDENTE, semelhante a um pé de galinha, representa o lado N do relacionamento entre as tabelas.

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

ESQUEMA TEXTUAL DE BANCO DE DADOS RELACIONAL

O banco de dados mostrado no diagrama anterior, pode ser declarado sob o formato textual, conforme exemplo a seguir:

FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO, DTNASCIMENTO)
DEPENDENTE (CODIGOFUNCIONARIO, NRDEPENDENTE, NOME, DTNASCIMENTO)
CODIGOFUNCIONARIO REFERENCIA FUNCIONARIO

Observe que, no esquema textual, as tabelas são declaradas com informações sobre o nome, além de uma lista contendo as suas respectivas colunas. As chaves primárias são sublinhadas. Por fim, as chaves estrangeiras são declaradas usando o padrão *nomecoluna(s) referencia nometabela(s)* .

Neste módulo, estudamos os principais elementos do modelo relacional de banco de dados.

VERIFICANDO O APRENDIZADO

1. A RESPEITO DOS TIPOS DE CHAVES EM PROJETO DE BANCO DE DADOS RELACIONAL, ANALISE AS PROPOSIÇÕES A SEGUIR:

I. QUANDO MAIS DE UMA COLUNA SERVIR PARA DIFERENCIAR UMA LINHA DAS DEMAIS EM UMA TABELA RELACIONAL E UMA DELAS É ESCOLHIDA COMO CHAVE PRIMÁRIA, AS RESTANTES SÃO DENOMINADAS CHAVES ALTERNATIVAS.
II. UMA CHAVE PRIMÁRIA CORRESPONDE A UMA COLUNA OU UMA COMBINAÇÃO DE COLUNAS CUJOS VALORES SERVEM PARA DIFERENCIAR UMA LINHA DAS DEMAIS DE UMA TABELA.
III. A CHAVE ESTRANGEIRA PERMITE A IMPLEMENTAÇÃO DE RELACIONAMENTOS EM UM BANCO DE DADOS RELACIONAL.

ASSINALE A ALTERNATIVA VERDADEIRA

A) Somente a proposição I é correta.

B) Somente a proposição II é correta.

C) Somente a proposição IIII é correta.

D) Todas as proposições são corretas.

2. CONSIDERE QUE EM UMA INSTITUIÇÃO DE ENSINO SUPERIOR (IES) EXISTA UM BANCO DE DADOS RELACIONAL DENOMINADO BDIES QUE POSSUI A TABELA ALUNO CUJOS CAMPOS ESTÃO ASSIM DESCRITOS:

ALUNO		
CODIGOALUNO	INT	PK
NOME	CHAR(90)	
SEXO	CHAR(1)	
DTNASCIMENTO	DATE	

ATENÇÃO! PARA VISUALIZAÇÃO COMPLETA DA TABELA UTILIZE A ROLAGEM HORIZONTAL

DE ACORDO COM OS FUNDAMENTOS DO MODELO RELACIONAL, É CORRETO AFIRMAR QUE:

A) O banco de dados BDIES pode possuir uma coleção de tabelas, todas com nome exclusivo, com colunas que podem conter valores dentro de um domínio.

B) A coluna CODIGOALUNO é chave estrangeira na tabela ALUNO.

C) A coluna CODIGOALUNO pode ser opcional.

D) A coluna NOME é multivalorada.

GABARITO

1. A respeito dos tipos de chaves em projeto de banco de dados relacional, analise as proposições a seguir:

I. Quando mais de uma coluna servir para diferenciar uma linha das demais em uma tabela relacional e uma delas é escolhida como chave primária, as restantes são denominadas chaves alternativas.
II. Uma chave primária corresponde a uma coluna ou uma combinação de colunas cujos valores servem para diferenciar uma linha das demais de uma tabela.
III. A chave estrangeira permite a implementação de relacionamentos em um banco de dados relacional.

Assinale a alternativa verdadeira

A alternativa **"D "** está correta.

De fato, i) quando diversas colunas são candidatas à chave primária e uma é escolhida, as demais passam a ser denominadas chaves alternativas; ii) o objetivo da chave primária é tornar única cada linha de uma tabela do banco de dados; iii) as chaves estrangeiras representam restrições capazes de representar relacionamentos entre tabelas.

2. Considere que em uma instituição de ensino superior (IES) exista um banco de dados relacional denominado BDIES que possui a tabela ALUNO cujos campos estão assim descritos:

ALUNO		
CODIGOALUNO	**int**	**PK**
NOME	**char(90)**	
SEXO	**char(1)**	
DTNASCIMENTO	**date**	

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

De acordo com os fundamentos do modelo relacional, é correto afirmar que:

A alternativa **"A "** está correta.

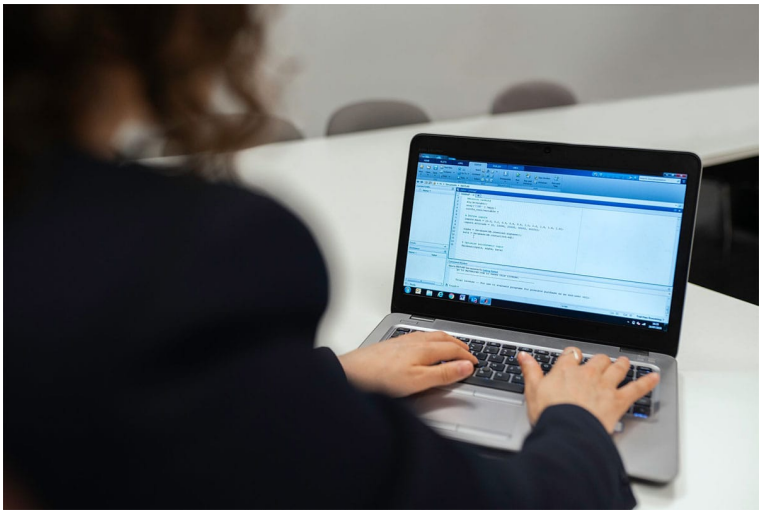
De fato, toda tabela de um banco de dados relacional tem que possuir um nome único. As colunas de uma tabela devem estar associadas a um domínio ou tipo de dados.

MÓDULO 2

⦿ Diferenciar formas normais

NORMALIZAÇÃO

Ao longo da nossa jornada, conhecemos os principais elementos do modelo relacional. Quando trabalhamos com modelagem das tabelas de um banco de dados, ao criarmos as tabelas, é natural definirmos colunas que têm relação com as características do objeto sendo modelado.



Fonte: ThisisEngineering/Unsplash

No entanto, modelar um banco de dados relacional não se resume simplesmente a usar uma **ferramenta CASE** e adicionar tabelas e relacionamentos sem que haja algum critério para essa construção.

FERRAMENTA CASE

(de Computer-Aided Software Engineering – Engenharia de Software Apoiada por Computador): software de apoio ao desenvolvimento de sistemas, desde a análise e modelagem até a programação e testes.

Ao longo deste módulo, estudaremos o assunto **normalização**, que ajudará a responder se um banco de dados foi bem projetado. Além disso, é possível executar o processo de normalização a partir de qualquer representação de dados. Isso significa que podemos iniciar o processo a partir de uma tela de sistema, ou mesmo um relatório.

A normalização é um processo baseado no conceito de forma normal (FN), que pode ser vista como uma regra, a qual deve ser observada na semântica de uma tabela, para que a considerem bem projetada.

📢 ATENÇÃO

Na literatura de banco de dados, há diversas formas normais: 1FN, 2FN, 3FN, FNBC, 4FN e 5FN. No entanto, para fins práticos, no contexto da maior parte dos projetos de banco de dados relacionais, costumamos executar o processo de normalização até a 3FN.

Dividiremos o processo de normalização até a 3FN de acordo com o seguinte roteiro:

Identificar a origem dos dados;
Construir tabela não normalizada a partir dos dados;
Aplicar as regras da primeira forma normal (1FN);

Aplicar as regras da segunda forma normal (2FN);
Aplicar as regras da terceira forma normal (3FN).

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

VAMOS ESTUDAR UM EXEMPLO?

Considere o relatório expresso na figura, que informa os docentes participantes em projetos de pesquisa de uma instituição de ensino superior (IES):

RELATÓRIOS DE ALOCAÇÃO DOCENTE A PROJETOS DE PESQUISA

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

CÓDIGO DO PROJETO: PRODATA DESCRIÇÃO: DESENVOLVIMENTO DE AMBIENTE PARA ANÁLISE DE DADOS			TIPO: ANÁLISE DE DADOS		
CÓDIGO DO DOCENTE	NOME	CATEGORIA	SALÁRIO	DATA DE INÍCIO	TEMPO ALOCADO
DOC001	JOSÉ	ADJUNTO	R$ 6.000,00	01/02/2019	16
DOC002	LUCIANO	TITULAR	R$ 16.000,00	01/02/2020	4
DOC003	GILSON	ADJUNTO	R$ 6.000,00	01/02/2019	16
DOC004	MARTA	TITULAR	R$ 16.000,00	01/02/2020	4
CÓDIGO DO PROJETO: PROMED DESCRIÇÃO: ATENDIMENTO COMUNITÁRIO E VACINAÇÃO			TIPO: ANÁLISE CLÍNICA		
CÓDIGO DO DOCENTE	NOME	CATEGORIA	SALÁRIO	DATA DE INÍCIO	TEMPO ALOCADO
DOC001	JOSÉ	ADJUNTO	R$ 6.000,00	01/02/2019	16

DOC010	MARIA	ADJUNTO	R$ 6.000,00	01/06/2020	0
DOC004	MARTA	TITULAR	R$ 16.000,00	01/05/2020	1

Figura: Alocação de docentes a projetos de pesquisa em uma IES.

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

De acordo com o relatório, nós podemos perceber que:

Os projetos são caracterizados por código, descrição e categoria (tipo).
Para cada docente alocado em projeto, aparecem os seguintes campos: código e nome do docente, sua categoria e salário, além da data de início de atuação no projeto, bem como o tempo alocado.

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

Agora nós executaremos o segundo passo do roteiro, que corresponde a criar tabela não normalizada a partir do relatório.

TABELA NÃO NORMALIZADA

A figura a seguir representa uma tabela não normalizada, denominada PROJETO, criada a partir do relatório:

CODIGOPROJETO	TIPO	DESCRICAO	DOCENTE				
			CODIGODOCENTE	NOME	CATEGORIA	SALARIO	DATAINIC
PRODATA	ANÁLISE DE DADOS	DESENVOLVIMENTO DE AMBIENTE PARA ANÁLISE DE DADOS	DOC001	JOSÉ	ADJUNTO	R$ 6.000,00	01/02/201
			DOC002	LUCIANO	TITULAR	R$ 16.000,00	01/02/202
			DOC003	GILSON	ADJUNTO	R$ 6.000,00	01/02/201
			DOC004	MARTA	TITULAR	R$ 16.000,00	01/02/202
PROMED	ANÁLISE CLÍNICA	ATENDIMENTO COMUNITÁRIO E VACINAÇÃO	DOC001	JOSÉ	ADJUNTO	R$ 6.000,00	01/02/201
			DOC010	MARIA	ADJUNTO	R$ 6.000,00	01/06/202
			DOC004	MARTA	TITULAR	R$ 16.000,00	01/05/202

Figura: Representação dos dados do relatório em tabela não normalizada.

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

A representação textual da tabela está expressa a seguir:

PROJETO (CODIGOPROJETO, TIPO, DESCRICAO, (CODIGODOCENTE, NOME, CATEGORIA, SALARIO, DATAINICIO, TEMPOMESES))

Nessa representação, a coluna CODIGOPROJETO diferencia cada projeto dos demais. A coluna CODIGODOCENTE diferencia os docentes alocados no contexto de um projeto.

Agora, observe com atenção a figura representativa da tabela não normalizada:

Cada linha da tabela não normalizada representa a informação de alocação de um docente a um projeto de pesquisa;
Note que a coluna DOCENTE é composta por um conjunto de colunas: CODIGODOCENTE, NOME, CATEGORIA, SALARIO, DATAINICIO e TEMPOMESES. Estamos diante de uma coluna composta;
Perceba, também, que se olharmos isoladamente para os valores das colunas que compõem a coluna DOCENTE, vamos perceber repetição. Isso acontece no caso dos funcionários José e Marta. A mesma informação está representada mais de uma vez, o que representa redundância.

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

Agora que construímos a tabela não normalizada, vamos à próxima etapa, que terá como saída um conjunto de tabelas na 1FN.

PRIMEIRA FORMA NORMAL (1FN)

Uma tabela está de acordo com a 1FN quando não possui atributo(s) multivalorado(s) nem atributo(s) composto(s).

Devemos recordar que, como resultado da etapa anterior, foi criada tabela não normalizada. Para ficar de acordo com a 1FN, nós executaremos os passos a seguir:

Criar tabela na 1FN com a mesma chave primária da tabela não normalizada, além das colunas atômicas da própria tabela não normalizada.

Criar uma tabela na 1FN para cada coluna composta, identificada na tabela não normalizada. Cada tabela terá uma chave primária composta pela chave primária da tabela criada no passo anterior e pela coluna identificada como composta. Além disso, terá as colunas membro da coluna em questão.

Criar uma tabela na 1FN para cada coluna multivalorada. Cada tabela terá uma chave primária composta pela chave primária da tabela não normalizada e pela coluna identificada como multivalorada.

Ao aplicarmos os passos à tabela PROJETO, teremos como resultado as seguintes tabelas em 1FN:

PROJETO (CODIGOPROJETO, TIPO, DESCRICAO)
PROJETODOCENTE (CODIGOPROJETO, CODIGODOCENTE, NOME, CATEGORIA, SALARIO, DATAINICIO, TEMPOMESES)

Ainda, de acordo com a representação textual, devemos perceber que:

A coluna CODIGOPROJETO da tabela PROJETO diferencia um projeto dos demais.

As colunas CODIGOPROJETO, CODIGODOCENTE compõem a chave primária da tabela PROJETODOCENTE, visto que um mesmo docente pode atuar em diversos projetos.

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

A figura a seguir representa o conteúdo das tabelas, com base nos dados originalmente expressos no relatório de alocação docente a projetos:

PROJETO

CODIGOPROJETO	TIPO	DESCRICAO
PRODATA	ANÁLISE DE DADOS	DESENVOLVIMENTO DE AMBIENTE PARA ANÁLISE DE DADOS
PROMED	ANÁLISE CLÍNICA	ATENDIMENTO COMUNITÁRIO E VACINAÇÃO

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

PROJETODOCENTE

CODIGOPROJETO	CODIGODOCENTE	NOME	CATEGORIA	SALARIO	DATAINICIO	TEMPOMESES
PRODATA	DOC001	JOSÉ	ADJUNTO	R$ 6.000,00	01/02/2019	16
PRODATA	DOC002	LUCIANO	TITULAR	R$ 16.000,00	01/02/2020	4
PRODATA	DOC003	GILSON	ADJUNTO	R$ 6.000,00	01/02/2019	16
PRODATA	DOC004	MARTA	TITULAR	R$ 16.000,00	01/02/2020	4
PROMED	DOC001	JOSÉ	ADJUNTO	R$ 6.000,00	01/02/2019	16
PROMED	DOC010	MARIA	ADJUNTO	R$ 6.000,00	01/06/2020	0
PROMED	DOC004	MARTA	TITULAR	R$ 16.000,00	01/05/2020	1

Figura: Representação dos dados do relatório em tabelas na 1FN.

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

DEPENDÊNCIA FUNCIONAL

Se observarmos os dados da tabela PROJETODOCENTE, vamos concluir que o nome do docente é o mesmo para cada código de docente. Parece então existir uma relação de dependência entre as colunas NOME e CODIGODOCENTE.

Assim, podemos expressar essa relação da seguinte maneira:

CODIGODOCENTE → NOME.

Com isso, dizemos que a coluna CODIGODOCENTE é determinante da coluna NOME.

Podemos dizer também que NOME é dependente de CODIGODOCENTE, isto é, determinado por CODIGODOCENTE.

📢 ATENÇÃO

A generalização dessa informação representa o conceito de dependência funcional.

Dessa forma, seja X um conjunto de atributos e Y um atributo; X → Y significa que o conjunto de atributos X determina o atributo Y.

DEPENDÊNCIA FUNCIONAL PARCIAL

Observe novamente a tabela PROJETODOCENTE expressa na figura:

PROJETODOCENTE

CODIGOPROJETO	CODIGODOCENTE	NOME	CATEGORIA	SALARIO	DATAINICIO	TEMPOMESES
PRODATA	DOC001	JOSÉ	ADJUNTO	R$ 6.000,00	01/02/2019	16
PRODATA	DOC002	LUCIANO	TITULAR	R$ 16.000,00	01/02/2020	4
PRODATA	DOC003	GILSON	ADJUNTO	R$ 6.000,00	01/02/2019	16
PRODATA	DOC004	MARTA	TITULAR	R$ 16.000,00	01/02/2020	4
PROMED	DOC001	JOSÉ	ADJUNTO	R$ 6.000,00	01/02/2019	16
PROMED	DOC010	MARIA	ADJUNTO	R$ 6.000,00	01/06/2020	0
PROMED	DOC004	MARTA	TITULAR	R$ 16.000,00	01/05/2020	1

📷 Figura: Representação dos dados da tabela PROJETODOCENTE.

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

Se analisarmos a relação CODIGOPROJETO, CODIGODOCENTE → NOME, iremos perceber que NOME é dependente somente de CODIGODOCENTE, ou seja, não é necessária a existência do par CODIGOPROJETO, CODIGODOCENTE para determinar o nome do docente. Estamos diante de uma dependência funcional parcial, visto que identificamos uma coluna dependente somente de parte da chave primária composta.

SEGUNDA FORMA NORMAL (2FN)

Uma tabela está na 2FN, caso esteja na 1FN e não haja dependências funcionais parciais.

Se analisarmos com um pouco mais de atenção cada coluna não chave da tabela PROJETODOCENTE, iremos perceber que, além da coluna NOME, as colunas CATEGORIA e SALARIO também só dependem da coluna CODIGODOCENTE.

Para ficar de acordo com a 2FN, será necessário eliminarmos as dependências parciais, conforme os passos a seguir:

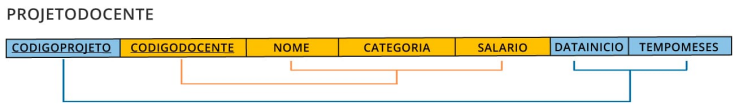
Manter no modelo cada tabela que possua chave primária simples;

⬇⬇

Identificar cada dependência parcial;

Criar uma tabela para cada dependência parcial identificada.

A figura a seguir identifica as colunas dependentes de parte da chave primária na tabela PROJETODOCENTE:



Fonte: Autor

Figura: Representação das dependências parciais na tabela PROJETODOCENTE.

Ao aplicarmos os passos ao modelo, teremos como resultado as seguintes tabelas na 2FN:

PROJETO (CODIGOPROJETO, TIPO, DESCRICAO)
PROJETODOCENTE (CODIGOPROJETO, CODIGODOCENTE, DATAINICIO,TEMPOMESES)
DOCENTE (CODIGODOCENTE, NOME, CATEGORIA, SALARIO)

A figura a seguir representa o conteúdo das tabelas na 2FN, com base nos dados, originalmente, expressos no relatório de alocação docente a projetos:

PROJETO

CODIGOPROJETO	TIPO	DESCRICAO
PRODATA	ANÁLISE DE DADOS	DESENVOLVIMENTO DE AMBIENTE PARA ANÁLISE
PROMED	ANÁLISE CLÍNICA	ATENDIMENTO COMUNITÁRIO E VACINAÇÃO

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

PROJETODOCENTE

CODIGOPROJETO	CODIGODOCENTE	DATAINICIO	TEMPOMESES
PRODATA	DOC001	01/02/2019	16
PRODATA	DOC002	01/02/2020	4
PRODATA	DOC003	01/02/2019	16
PRODATA	DOC004	01/02/2020	4
PROMED	DOC001	01/02/2019	16
PROMED	DOC010	01/06/2020	0
PROMED	DOC004	01/05/2020	1

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

DOCENTE

CODIGODOCENTE	NOME	CATEGORIA	SALARIO
DOC001	JOSÉ	ADJUNTO	R$ 6.000,00
DOC002	LUCIANO	TITULAR	R$ 16.000,00
DOC003	GILSON	ADJUNTO	R$ 6.000,00
DOC004	MARTA	TITULAR	R$ 16.000,00
DOC010	MARIA	ADJUNTO	R$ 6.000,00

Figura: Representação dos dados do relatório em tabelas em 2FN.

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

O modelo está na 2FN, pois, além de estar na 1FN, não existem dependências parciais.

DEPENDÊNCIA FUNCIONAL TRANSITIVA

Se observarmos os dados da tabela DOCENTE gerada na etapa anterior, podemos concluir que o valor do salário é o mesmo em cada categoria. Perceba que parece então existir uma relação de dependência entre as colunas CATEGORIA e SALARIO. Assim, podemos expressar essa relação da seguinte maneira:

CATEGORIA → SALARIO.

Com isso, dizemos que a coluna CATEGORIA é determinante da coluna SALARIO. Podemos dizer também que SALARIO é dependente de CATEGORIA.

ATENÇÃO

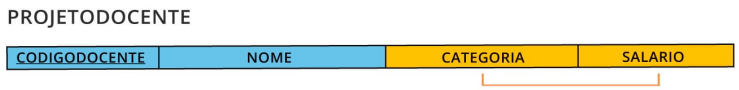
Estamos diante de um exemplo de dependência funcional, em que o determinante é uma coluna que não pertence à chave primária da tabela.

Uma dependência funcional transitiva ocorre quando uma coluna é dependente de alguma coluna não-chave da tabela.

TERCEIRA FORMA NORMAL (3FN)

Uma tabela está em 3FN caso esteja na 2FN e não possua dependências transitivas.

A figura a seguir identifica a dependência transitiva na tabela PROJETODOCENTE:



Fonte: Autor

Figura: Representação da dependência transitiva na tabela PROJETODOCENTE.

Para ficar de acordo com a 3FN, será necessário eliminarmos as dependências transitivas, conforme os passos a seguir:

Manter no modelo cada tabela que tenha menos de duas colunas não chave;

Identificar cada dependência transitiva;

Criar uma tabela para cada dependência transitiva identificada.

Ao aplicarmos os passos ao modelo, teremos como resultado as seguintes tabelas na 3FN:

PROJETO (CODIGOPROJETO, TIPO, DESCRICAO)
PROJETODOCENTE (CODIGOPROJETO, CODIGODOCENTE, DATAINICIO, TEMPOMESES)
DOCENTE (CODIGODOCENTE, NOME, CATEGORIA)
CATEGORIA (CATEGORIA, SALARIO)

A figura a seguir representa o conteúdo das tabelas na 3FN, com base nos dados originalmente expressos no relatório de alocação de docentes a projetos:

PROJETO

CODIGOPROJETO	TIPO	DESCRICAO
PRODATA	ANÁLISE DE DADOS	DESENVOLVIMENTO DE AMBIENTE PARA ANÁLISE
PROMED	ANÁLISE CLÍNICA	ATENDIMENTO COMUNITÁRIO E VACINAÇÃO

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

PROJETODOCENTE

CODIGOPROJETO	CODIGODOCENTE	DATAINICIO	TEMPOMESES
PRODATA	DOC001	01/02/2019	16
PRODATA	DOC002	01/02/2020	4
PRODATA	DOC003	01/02/2019	16
PRODATA	DOC004	01/02/2020	4
PROMED	DOC001	01/02/2019	16
PROMED	DOC010	01/06/2020	0
PROMED	DOC004	01/05/2020	1

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

DOCENTE

CODIGODOCENTE	NOME	CATEGORIA

DOC001	JOSÉ	ADJUNTO
DOC002	LUCIANO	TITULAR
DOC003	GILSON	ADJUNTO
DOC004	MARTA	TITULAR
DOC010	MARIA	ADJUNTO

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

CATEGORIA

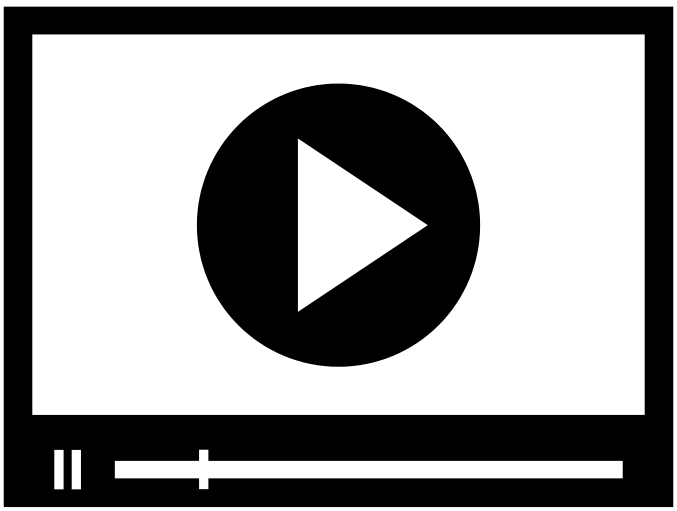
CATEGORIA	SALARIO
ADJUNTO	R$ 6.000,00
TITULAR	R$ 16.000,00

Figura: Representação dos dados do relatório em tabelas na 3FN.

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

O modelo está na 3FN, pois, além de estar na 2FN, não existem dependências transitivas.

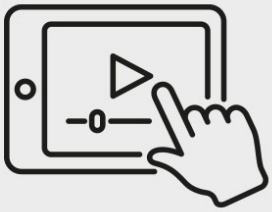
Neste módulo, nós aprendemos que o processo de normalização é útil para refinarmos a construção de um banco de dados relacional. Estudamos três formas normais e percebemos que, ao normalizarmos o nosso modelo, o número de tabelas tende a aumentar, minimizando assim redundância nos dados.



TABELAS NA 3FN: CONSEQUÊNCIA DE UMA BOA MODELAGEM CONCEITUAL

Vamos assistir ao vídeo a seguir:

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



VERIFICANDO O APRENDIZADO

1. O PROCESSO DE NORMALIZAÇÃO OBJETIVA ELIMINAR DADOS REDUNDANTES, ALÉM DE GARANTIR QUE A DEPENDÊNCIA DE DADOS FAÇA SENTIDO. A RESPEITO DA TERCEIRA FORMA NORMAL EM UMA TABELA RELACIONAL, ASSINALE A ALTERNATIVA VERDADEIRA:

A) A tabela não precisa estar na 1FN.

B) A tabela não precisa estar na 2FN.

C) A tabela não pode conter campos opcionais

D) A tabela precisa estar na 2FN e não deve possuir coluna(s) com dependência(s) transitiva(s).

2. SEJA UMA RELAÇÃO R (C1, C2, C3, C4, C5) COM CHAVE PRIMÁRIA COMPOSTA PELAS COLUNAS C1 E C2 E AS SEGUINTES DEPENDÊNCIAS FUNCIONAIS:

C1 → C3
C3 → C4
C2 → C5

PODEMOS AFIRMAR QUE O ESQUEMA R ATENDE À:

A) 1 FN

B) 2 FN

C) 3 FN

D) 4 FN

GABARITO

1. O processo de Normalização objetiva eliminar dados redundantes, além de garantir que a dependência de dados faça sentido. A respeito da terceira forma normal em uma tabela relacional, assinale a alternativa verdadeira:

A alternativa **"D "** está correta.

De fato, para estar na 3FN a tabela tem que respeitar as regras da 2FN, pois uma forma engloba as restrições da forma anterior. Além disso, não pode haver dependências entre colunas não-chave.

2. Seja uma relação R (C1, C2, C3, C4, C5) com chave primária composta pelas colunas C1 e C2 e as seguintes dependências funcionais:

C1 → C3
C3 → C4

C2 → C5

Podemos afirmar que o esquema R atende à:

A alternativa **"A "** está correta.

De fato, se C1 determina C3, temos então uma dependência parcial da chave primária, o que é proibido na 2FN. Como cada FN, a partir da segunda, depende da anterior, a relação R atende à 1FN.

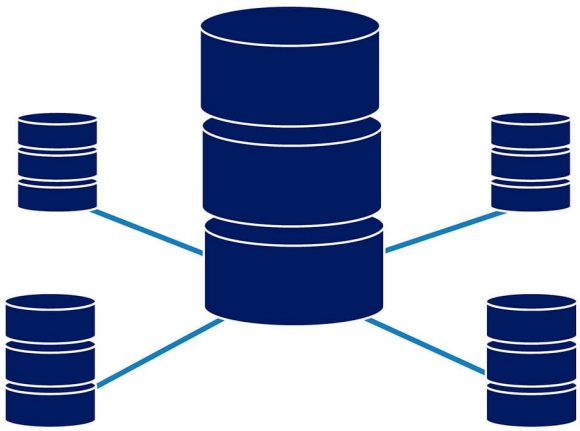
MÓDULO 3

⦿ Aplicar o mapeamento conceitual-lógico

MAPEAMENTO CONCEITUAL-LÓGICO

No projeto de banco de dados, há etapas que devem ser seguidas para a construção de um modelo conceitual. Para tanto, é usual a adoção do diagrama de entidade e relacionamento (DER) - tipo de modelo conceitual em que os principais objetos do negócio modelado se tornam entidades caracterizadas por atributos e relacionamentos.

Estudamos os principais componentes do modelo relacional, um modelo lógico que é base para a implementação de bancos de dados relacionais.



Fonte: Tumisu/Pixabay

Neste módulo, construiremos a ponte entre os modelos conceitual e lógico.

Passaremos, então, a aplicar regras formais de mapeamento do modelo conceitual, visando a construção do modelo lógico. Na prática, projetaremos um banco de dados relacional a partir de um DER.

REGRAS DE MAPEAMENTO

Podemos dividir o mapeamento conceitual-lógico em quatro etapas:

ENTIDADES

RELACIONAMENTOS

ATRIBUTOS MULTIVALORADOS

ESPECIALIZAÇÃO/GENERALIZAÇÃO

A seguir, conheceremos as regras de cada etapa do mapeamento e a sua respectiva aplicação por meio de exemplos.

MAPEAMENTO DE ENTIDADES

O mapeamento de entidades envolve:

ENTIDADE FORTE OU INDEPENDENTE

Cada entidade E vira uma tabela T;

Cada atributo simples da entidade E vira uma coluna na tabela T;

O atributo identificador da entidade E vira chave primária na tabela T.

ENTIDADE FRACA OU DEPENDENTE:

Cada entidade fraca F vira uma tabela T;

Cada atributo simples da entidade F vira uma coluna na tabela T;

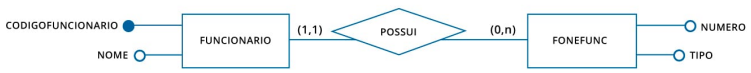
A tabela T possuirá chave estrangeira originada a partir da chave primária da tabela proprietária;

A tabela T possuirá chave primária composta pela coluna chave estrangeira criada no passo anterior e pela coluna mapeada do atributo identificador da entidade F na tabela T.

EXEMPLO DE MAPEAMENTO DE ENTIDADES:

Conhecidas as regras para o mapeamento de entidades, observe no exemplo a seguir um diagrama de entidade e relacionamento (DER) contendo duas entidades:

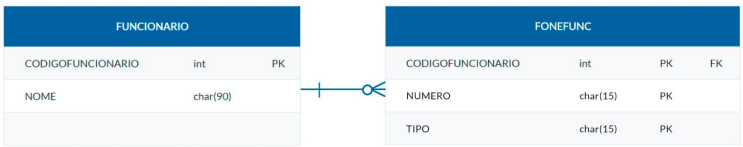
FUNCIONARIO (entidade forte) e FONEFUNC (entidade fraca).



Fonte: Autor

Figura: Diagrama de Entidade e Relacionamento (DER) contendo duas entidades.

A figura a seguir exibe o modelo lógico gerado:



Fonte: Autor

Figura: Tabelas criadas com base no mapeamento conceitual-lógico envolvendo **entidade**.

Note que a tabela FUNCIONARIO corresponde à aplicação da regra para entidade forte. De modo semelhante, a tabela FONEFUNC corresponde à aplicação da regra para entidade fraca.

Após a aplicação das regras de mapeamento, foi gerada a representação textual a seguir:

FUNCIONARIO (CODGOFUNCIONARIO, NOME)
FONEFUNC (CODIGOFUNCIONARIO, NUMERO, TIPO)
CODIGOFUNCIONARIO REFERENCIA FUNCIONARIO

MAPEAMENTO DE RELACIONAMENTOS

O mapeamento de relacionamentos dependerá da cardinalidade máxima:

RELACIONAMENTOS 1:1:

Cardinalidade (0,1):(0,1): priorizar adição de coluna(s). Alternativa: tabela própria.

Cardinalidade (0,1):(1,1): priorizar fusão de tabelas. Alternativa: adição de colunas.

Cardinalidade (1,1):(1,1): fusão de tabelas.

RELACIONAMENTOS 1:N:

Identificar a tabela T do lado N.

Adicionar chave estrangeira na tabela T do lado N referente à chave primária da tabela do lado 1.

Cada atributo simples do relacionamento vira uma coluna na tabela T.

RELACIONAMENTOS N:N:

Cada relacionamento vira uma tabela T.

A tabela T possuirá chaves estrangeiras originadas das chaves primárias das tabelas participantes do relacionamento.

A tabela T possuirá chave primária composta pelas chaves estrangeiras criadas no passo anterior.

Cada atributo simples do relacionamento vira uma coluna na tabela T.

RELACIONAMENTOS N-ÁRIOS (ANÁLOGO A RELACIONAMENTOS N:N):

Cada relacionamento vira uma tabela T.

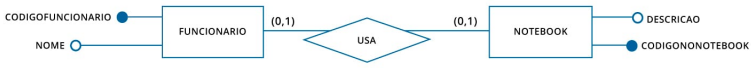
A tabela T possuirá chaves estrangeiras originadas das chaves primárias das tabelas participantes do relacionamento.

A tabela T possuirá chave primária composta pelas chaves estrangeiras criadas no passo anterior.

Cada atributo simples do relacionamento vira uma coluna na tabela T.

EXEMPLOS DE MAPEAMENTO DE RELACIONAMENTO 1:1:

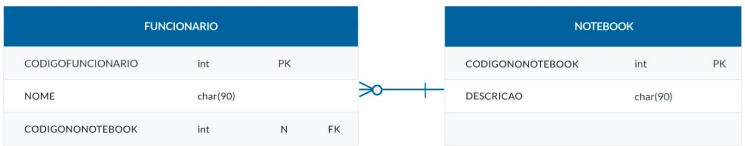
Conhecidas as regras para o mapeamento de relacionamentos 1:1, observe no exemplo a seguir um diagrama de entidade e relacionamento (DER) contendo duas entidades: FUNCIONARIO e NOTEBOOK. Há um relacionamento **1:1** no qual ambas as entidades possuem participação **opcional** (cardinalidades (0,1): (0,1)):



Fonte: Autor

📷 Figura: DER contendo relacionamento **1:1** – ambas as entidades com participação **opcional**.

A figura a seguir exibe o modelo lógico gerado:



Fonte: Autor

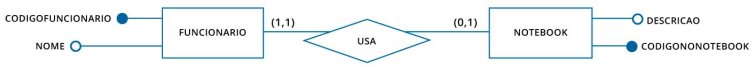
📷 Figura: Tabelas criadas com base no mapeamento conceitual-lógico envolvendo relacionamento **1:1** - ambas as entidades com participação **opcional**.

Para esse tipo de relacionamento, o mais adequado é priorizar adição de coluna(s), o que ocorreu ao criarmos a coluna CODIGONOTEBOOK como chave estrangeira na tabela FUNCIONARIO. Vale lembrar que estamos diante de uma coluna opcional.

A seguir, a representação textual do modelo:

NOTEBOOK (CODIGONOTEBOOK, DESCRICAO)
FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CODIGONOTEBOOK)
CODIGONOTEBOOK REFERENCIA NOTEBOOK

A figura a seguir apresenta um DER contendo um relacionamento **1:1** no qual há uma entidade com participação obrigatória e a outra **opcional** (cardinalidades (0,1): (1,1)):



Fonte: Autor

📷 Figura DER contendo relacionamento **1:1** – uma entidade com participação **obrigatória** e a outra **opcional**.

A figura a seguir exibe o modelo lógico gerado:

FUNCIONARIO		
CODIGOFUNCIONARIO	int	PK
NOME	char(90)	
CODIGONONOTEBOOK	int	N
DESCRICAO	char(90)	N

Figura: Tabela criada com base no mapeamento conceitual-lógico envolvendo relacionamento **1:1** - uma entidade com participação **obrigatória** e a outra **opcional**.

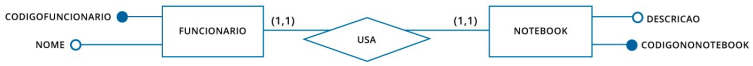
Atenção! Para visualização completa da tabela utilize a rolagem horizontal

Para esse tipo de relacionamento, o mais adequado é priorizar fusão de tabelas, o que ocorreu ao criarmos as colunas CODIGONOTEBOOK e DESCRICAO na tabela FUNCIONARIO, ambas opcionais.

Após a aplicação das regras de mapeamento, foi gerada a representação textual a seguir:

FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CODIGONOTEBOOK, DESCRICAO)

A figura a seguir apresenta um DER parcial contendo um relacionamento **1:1** e ambas as entidades com participação obrigatória (cardinalidades (1,1): (1,1)):



Fonte: Autor

Figura: DER parcial contendo relacionamento **1:1** – ambas as entidades com participação obrigatória.

A figura a seguir exibe o modelo lógico gerado:

FUNCIONARIO		
CODIGOFUNCIONARIO	int	PK
NOME	char(90)	
CODIGONONOTEBOOK	int	
DESCRICAO	char(90)	

Figura: Tabelas criadas com base no mapeamento conceitual-lógico envolvendo relacionamento **1:1** – ambas as entidades com participação obrigatória.

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

Para esse tipo de relacionamento, o mais adequado é priorizar fusão de tabelas, o que ocorreu ao criarmos as colunas CODIGONOTEBOOK e DESCRICAO na tabela FUNCIONARIO, ambas obrigatórias.

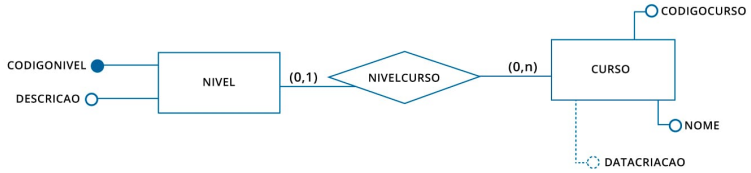
Após a aplicação das regras de mapeamento, foi gerada a representação textual a seguir:

FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CODIGONOTEBOOK, DESCRICAO)

EXEMPLO DE MAPEAMENTO DE RELACIONAMENTO 1:N:

Conhecidas as regras para o mapeamento de relacionamentos 1:N, observe no exemplo a seguir um diagrama de entidade e relacionamento (DER) contendo duas entidades: NIVEL e CURSO.

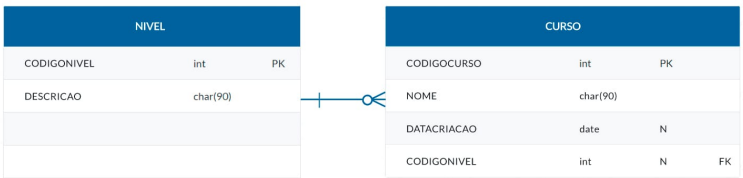
A figura a seguir mostra um DER parcial contendo um relacionamento **1:N**:



Fonte: Autor

Figura: DER contendo relacionamento **1:N**.

A figura a seguir exibe o modelo lógico gerado:



Fonte: Autor

Figura: Tabelas criadas com base no mapeamento conceitual-lógico envolvendo relacionamento **1:N**.

Para esse tipo de relacionamento, foi utilizada adição de coluna(s) na tabela do lado N, o que ocorreu ao criarmos a chave estrangeira CODIGONIVEL na tabela CURSO.

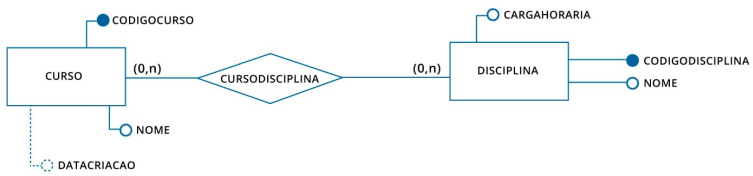
Após a aplicação das regras de mapeamento, foi gerada a representação textual a seguir:

NIVEL (CODGONIVEL, DESCRICAO)
CURSO (CODIGOCURSO, NOME, DATACRIACAO, CODIGONIVEL)
CODIGONIVEL REFERENCIA NIVEL

EXEMPLO DE MAPEAMENTO DE RELACIONAMENTO N:N:

Conhecidas as regras para o mapeamento de relacionamentos N:N, observe no exemplo a seguir um diagrama de entidade e relacionamento (DER) contendo duas entidades: CURSO e DISCIPLINA.

A figura a seguir mostra um DER parcial contendo um relacionamento **N:N**:



Fonte: Autor

Figura: DER parcial contendo relacionamento **N:N**.

A figura a seguir exibe o modelo lógico gerado:

Figura: Tabelas criadas com base no mapeamento conceitual-lógico envolvendo relacionamento **N:N**

Para esse tipo de relacionamento, foi utilizada tabela própria, o que ocorreu ao criarmos CURSODISCIPLINA, contendo duas chaves estrangeiras: CODIGOCURSO e CODIGODISCIPLINA. Ao mesmo tempo, a combinação das duas colunas forma a chave primária composta da tabela.

Após a aplicação das regras de mapeamento, foi gerada a representação textual a seguir:

CURSO (CODIGOCURSO, NOME, DATACRIACAO, CODIGONIVEL)
CODIGONIVEL REFERENCIA NIVEL
DISCIPLINA (CODIGODISCIPLINA, NOME, CARGAGORARIA)

CURSODISCIPLINA (CODIGOCURSO, CODIGODISCIPLINA)
CODIGOCURSO REFERENCIA CURSO
CODIGODISCIPLINA REFERENCIA DISCIPLINA

OBSERVAÇÃO SOBRE ENTIDADE ASSOCIATIVA

O mapeamento de entidade associativa, isto é, relacionamento com atributos, segue as regras utilizadas no mapeamento de relacionamentos N:N.

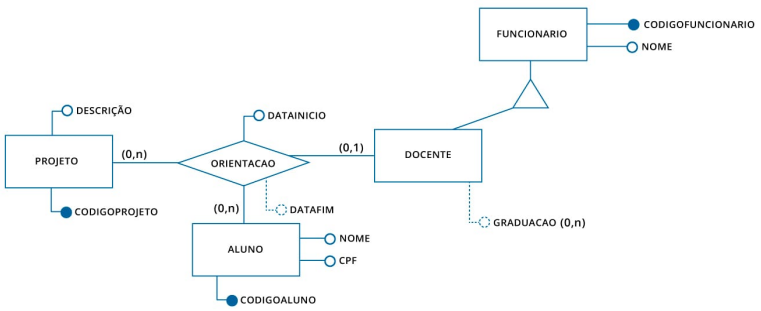
OBSERVAÇÃO SOBRE AUTORRELACIONAMENTO

O mapeamento de autorrelacionamento segue as regras utilizadas no mapeamento de relacionamentos. Basta, então, você ficar atento ao tipo de cardinalidade máxima em questão.

EXEMPLO DE MAPEAMENTO DE RELACIONAMENTO TERNÁRIO:

Conhecidas as regras para o mapeamento de relacionamentos, ao nos depararmos com relacionamentos ternários, precisaremos avaliar as cardinalidades máximas em questão. Observe no exemplo a seguir um diagrama de entidade e relacionamento (DER) contendo um relacionamento ternário entre as entidades PROJETO, DOCENTE e ALUNO.

A figura a seguir mostra um DER parcial contendo um relacionamento **ternário**.



Fonte: Autor

Figura DER parcial contendo relacionamento **ternário**.

A figura a seguir exibe o modelo lógico gerado:

Figura: Tabelas criadas com base no mapeamento conceitual-lógico envolvendo relacionamento **ternário**.

Para esse tipo de relacionamento, foi utilizada tabela própria, o que ocorreu ao criarmos ORIENTACAO, contendo três chaves estrangeiras: CODIGOFUNCIONARIO, CODIGOALUNO e CODIGOPROJETO. Além disso, foram criadas as colunas DATAINICIO e DATAFIM, as quais representam informações importantes sob o contexto de um registro de orientação.

Após a aplicação das regras de mapeamento, foi gerada a representação textual a seguir:

PROJETO (CODIGOPROJETO, DESCRICAO)
DOCENTE (CODIGOFUNCIONARIO, NOME)
ALUNO (CODIGOALUNO, NOME, CPF)
ORIENTACAO (CODIGOPROJETO, CODIGOALUNO, CODIGOFUNCIONARIO, DATAINICIO, DATAFIM)
CODIGOPROJETO REFERENCIA PROJETO

CODIGOALUNO REFERENCIA ALUNO

CODIGOFUNCIONARIO REFERENCIA DOCENTE

MAPEAMENTO DE ATRIBUTOS MULTIVALORADOS

O mapeamento de atributos multivalorados envolve:

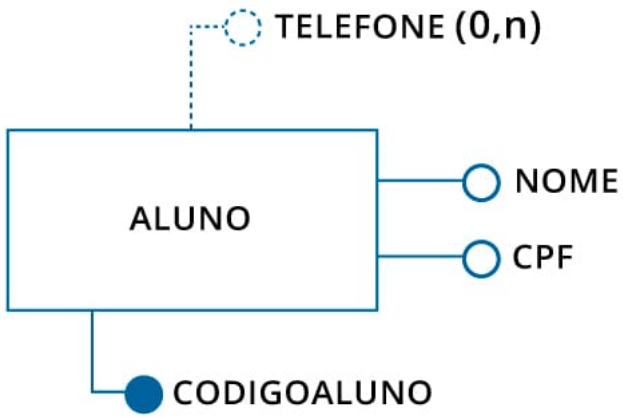
Criar uma tabela T para cada atributo multivalorado.

Criar coluna(s) para o(s) atributo(s) multivalorado(s).

A tabela T possuirá chave estrangeira originada da chave primária da tabela original.

A tabela T possuirá chave primária composta pela chave estrangeira criada no passo anterior e pela(s) coluna(s) referente(s) ao(s) atributo multivalorado(s).

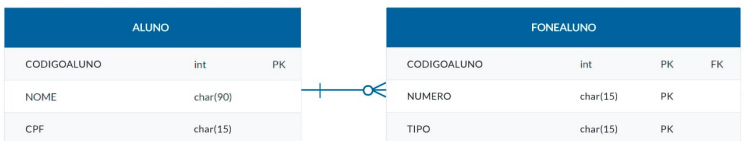
A figura a seguir mostra um DER parcial contendo **atributo multivalorado**.



Fonte: Autor

Figura DER parcial contendo atributo multivalorado.

A figura a seguir exibe o modelo lógico gerado:



Fonte: Autor

Figura: Tabelas criadas com base no mapeamento conceitual-lógico envolvendo atributo **multivalorado**.

Note que para mapear o atributo multivalorado TELEFONE, foi criada tabela própria denominada FONEALUNO. A coluna CODIGOALUNO de FONEALUNO é chave estrangeira. Além disso, as colunas CODIGOALUNO, NUMERO, TIPO representam uma chave primária composta. A coluna TIPO, criada na tabela FONEALUNO, representa a categoria do telefone, por exemplo, residencial, comercial ou móvel.

Após a aplicação das regras de mapeamento, foi gerada a representação textual a seguir:

ALUNO (CODIGOALUNO, NOME, CPF)
FONEALUNO (CODIGOALUNO, NUMETO, TIPO)
CODIGOALUNO REFERENCIA ALUNO

MAPEAMENTO DE ESPECIALIZAÇÃO/GENERALIZAÇÃO

O mapeamento de especialização/generalização envolve:

SOLUÇÃO I

Tabela única.

Criar uma tabela única que contenha todos os atributos das entidades genérica e especializadas.

Criar uma coluna TIPO, caso não exista, para identificar a entidade especializada.

SOLUÇÃO II

Uma tabela para cada entidade (genérica ou especializada) que compõe a hierarquia.

Criar uma tabela para a entidade genérica, com coluna(s) referente(s) ao(s) atributo(s) da entidade genérica.

Criar uma tabela para cada entidade especializada, com coluna(s) referentes ao(s) atributo(s) da entidade especializada.

Cada tabela de entidade especializada possuirá chave estrangeira originada da chave primária da tabela da entidade genérica. Esta também será a sua chave primária.

SOLUÇÃO III

Subdivisão da entidade genérica.

Nesta implementação, não será criada tabela para a entidade genérica.

Criar uma tabela para cada entidade especializada, com coluna(s) referentes ao(s) atributo(s) da entidade especializada.

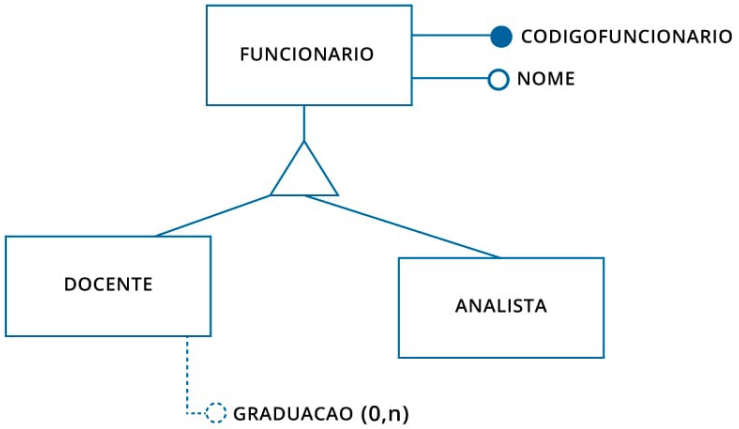
Em cada tabela, criar colunas referentes aos atributos da entidade genérica, sendo sua chave primária originada do atributo identificador da entidade genérica.

Caso a hierarquia seja parcial, será necessário criar uma tabela adicional para abarcar as entidades genéricas que não estão nas entidades especializadas. Essa tabela conterá somente os atributos da entidade genérica.

Importante notar que esta solução pode gerar redundância de dados, no caso de hierarquia sobreposta, requerendo um tratamento adicional para controle de redundância.

EXEMPLOS DE MAPEAMENTO DE ESPECIALIZAÇÃO/GENERALIZAÇÃO:

Conhecidas as regras para o mapeamento de **especialização/generalização**, observe no exemplo a seguir um diagrama de entidade e relacionamento (DER) com esse mecanismo:



Fonte: Autor

Figura: DER contendo especialização/generalização.

A figura a seguir exibe o modelo lógico gerado após a aplicação da **solução I** expressa nas regras de mapeamento:

FUNCIONARIO		
CODIGOFUNCIONARIO	int	PK
NOME	char(90)	
TIPO	char(40)	

Figura: Tabela criada com base no mapeamento conceitual-lógico envolvendo especialização/generalização – **solução I**.

Atenção! Para visualização completa da tabela utilize a rolagem horizontal

No exemplo, foi criada tabela única (FUNCIONARIO) contendo colunas referentes à entidade genérica, além da coluna TIPO, para identificar a categoria de cada colaborador.

Após a aplicação das regras de mapeamento, foi gerada a representação textual a seguir:

FUNCIONARIO (CODIGOFUNCIONARIO, NOME, TIPO)

A figura a seguir exibe o modelo lógico gerado após a aplicação da **solução II** das regras de mapeamento:

Figura: Tabelas criadas com base no mapeamento conceitual-lógico envolvendo especialização/generalização – **solução II**.

No exemplo, foram criadas três tabelas: uma (FUNCIONARIO) referente à entidade genérica. As restantes, DOCENTE e ANALISTA, referentes às entidades especializadas em questão. Cada CODIGOFUNCIONARIO presente nas tabelas DOCENTE e ANALISTA exerce o papel de chave estrangeira.

Após a aplicação das regras de mapeamento, foi gerada a representação textual a seguir:

FUNCIONARIO (CODIGOFUNCIONARIO, NOME)
DOCENTE (CODIGOFUNCIONARIO)
CODIGOFUNCIONARIO REFERENCIA FUNCIONARIO
ANALISTA (CODIGOFUNCIONARIO)
CODIGOFUNCIONARIO REFERENCIA FUNCIONARIO

A figura a seguir exibe o modelo lógico gerado após a aplicação da **solução III** das regras de mapeamento:

DOCENTE		
CODIGOFUNCIONARIO	int	PK
NOME	char(90)	

ANALISTA		
CODIGOFUNCIONARIO	int	PK
NOME	char(90)	

OUTROSFUNCIONARIOS		
CODIGOFUNCIONARIO	int	PK
NOME	char(90)	

Figura: Tabelas criadas com base no mapeamento conceitual-lógico envolvendo especialização/generalização – **solução III**.

No exemplo, foram criadas três tabelas, sendo OUTROSFUNCIONARIOS onde devem ser mantidos funcionários que não sejam docentes nem analistas. As restantes, DOCENTE e ANALISTA, são referentes às entidades especializadas em questão.

Após a aplicação das regras de mapeamento, foi gerada a representação textual a seguir:

DOCENTE (CODIGOFUNCIONARIO, NOME)
ANALISTA (CODIGOFUNCIONARIO, NOME)
OUTROSFUNCIONARIOS (CODIGOFUNCIONARIO, NOME)

Convém ressaltar que a tabela OUTROSFUNCIONARIOS seria necessária somente se o mecanismo de generalização/especialização fosse parcial, ou seja, se houvesse algum colaborador não enquadrado nas categorias docente ou analista.

ATENÇÃO

Como dica prática, devemos ter cuidado especial caso seja escolhida a **solução III**. Ao incluir um novo funcionário, será necessário verificar todas as tabelas criadas para as especializações para garantir a unicidade da chave primária. No exemplo, é preciso verificar os valores de chave primária nas tabelas DOCENTE, ANALISTA e OUTROSFUNCIONARIOS.

Por fim, convém ressaltar que a solução II é a mais usual, por ser mais flexível, dada a facilidade existente em contemplar novas especializações. Além disso, a solução I tende a gerar diversas ocorrências de valores nulos em colunas. Ao mesmo tempo, a solução III apresenta maior possibilidade de gerar redundância de dados.

ESTUDO DE CASO

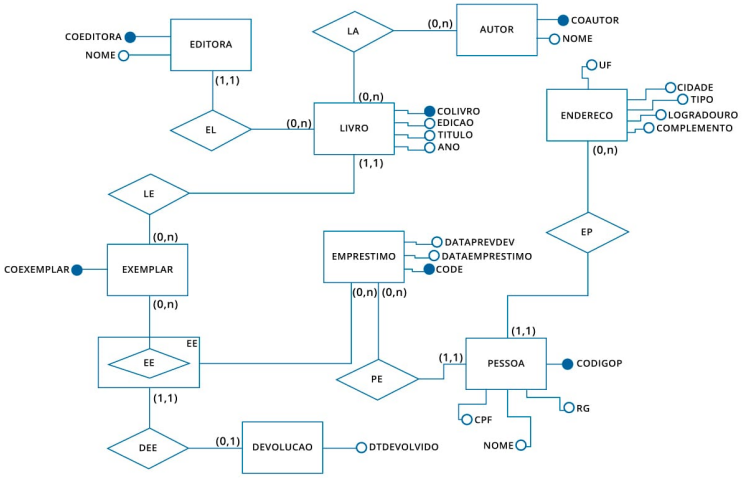
Para praticar as principais regras de mapeamento estudadas, vamos realizar um estudo de caso. Tendo como base os requisitos a seguir, foi construído o diagrama de entidade e relacionamento (DER) da figura abaixo. A partir desse DER, realize o mapeamento conceitual-lógico, sob a forma de descrição textual.

Deseja-se informatizar os processos de empréstimo e devolução de livros de uma biblioteca da rede pública municipal:

Cada pessoa habilitada a emprestar livros é identificada por um código. Além disso, são armazenados o RG, o CPF e o nome. Uma pessoa pode ter diversos endereços cadastrados, inclusive, nenhum. Cada endereço possui complemento, logradouro, UF, cidade e um tipo que o identifica.
Cada livro possui um código, ano, título e edição. Além disso, um livro pode ter vários autores. Cada autor é identificado por um código e possui um nome. Um livro pode ser inicialmente cadastrado sem a informação do autor. Um livro pode possuir vários exemplares, inclusive nenhum. Os exemplares serão emprestados.
Todo livro está associado a no máximo uma editora. Cada editora é identificada por um código e possui um nome.
Uma pessoa pode realizar diversos empréstimos, inclusive nenhum. Todo empréstimo possui um código que o identifica, além da data de empréstimo e data de previsão de devolução. Em um empréstimo a pessoa pode levar diversos exemplares. É necessário registrar a devolução de cada exemplar emprestado.

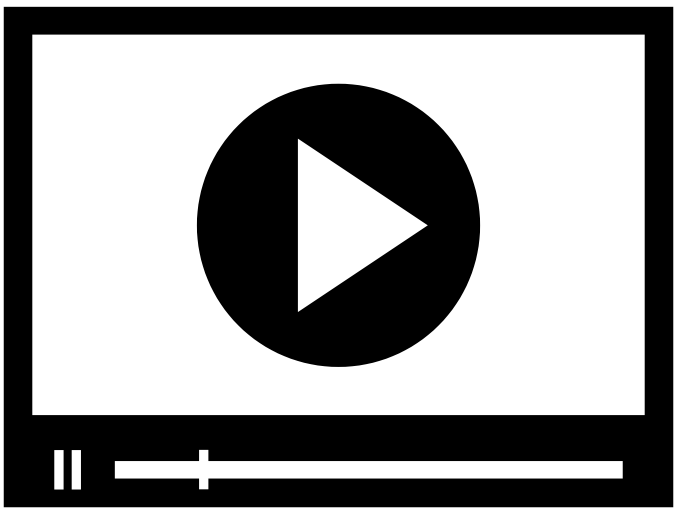
Atenção! Para visualização completa da tabela utilize a rolagem horizontal

Modelo de entidade e relacionamento resultante do projeto conceitual:



Fonte: Autor

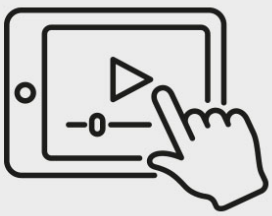
Figura: Modelo de entidade e relacionamento referente ao estudo de caso.



ESTUDO DE CASO

Assista, agora, ao desenvolvimento e conclusão do estudo de caso:

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



VERIFICANDO O APRENDIZADO

1. CONSIDERE UM RELACIONAMENTO ENTRE TURMA E DISCIPLINA (N:N) DENOMINADO OFERTA. PARA CADA DISCIPLINA OFERTADA POR UMA TURMA É NECESSÁRIO SABER O NÚMERO DE VAGAS. QUAL ALTERNATIVA A

SEGUIR REPRESENTA CORRETAMENTE O MODELO LÓGICO DERIVADO DO MODELO CONCEITUAL?

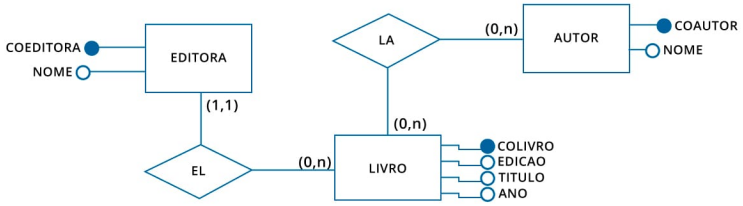
A) Somente uma tabela para todo o modelo.

B) Duas tabelas, uma para TURMA e outra para DISCIPLINA.

C) Três tabelas: uma para TURMA, uma para DISCIPLINA e uma para OFERTA. Além disso, o atributo que representa o número de vagas deve estar modelado na tabela OFERTA.

D) Três tabelas: uma para TURMA, uma para DISCIPLINA e uma para OFERTA.
Além disso, o atributo que representa o número de vagas deve estar modelado na tabela DISCIPLINA.

2. CONSIDERE O PROJETO CONCEITUAL PARCIAL A SEGUIR:



PARA CONSTRUIR O PROJETO LÓGICO A PARTIR DO DER, É CORRETO AFIRMAR QUE SERÃO CRIADAS:

A) Duas tabelas.

B) Três tabelas.

C) Quatro tabelas.

D) Cinco tabelas.

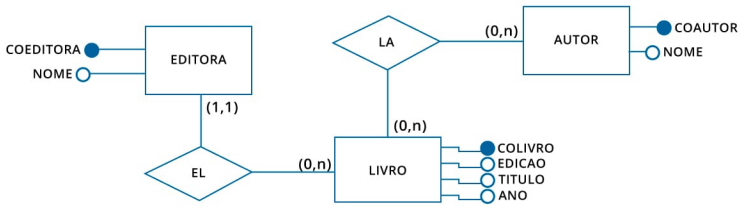
GABARITO

1. Considere um relacionamento entre TURMA e DISCIPLINA (N:N) denominado OFERTA. Para cada disciplina ofertada por uma turma é necessário saber o número de vagas. Qual alternativa a seguir representa corretamente o modelo lógico derivado do modelo conceitual?

A alternativa **"C "** está correta.

Diante da cardinalidade máxima N:N, a solução é criar três tabelas: duas para as entidades participantes e uma para o relacionamento. O atributo que representa o número de vagas, pelo fato de fazer parte do relacionamento, deve ser modelado na tabela OFERTA.

2. Considere o projeto conceitual parcial a seguir:



Para construir o projeto lógico a partir do DER, é correto afirmar que serão criadas:

A alternativa **"C "** está correta.

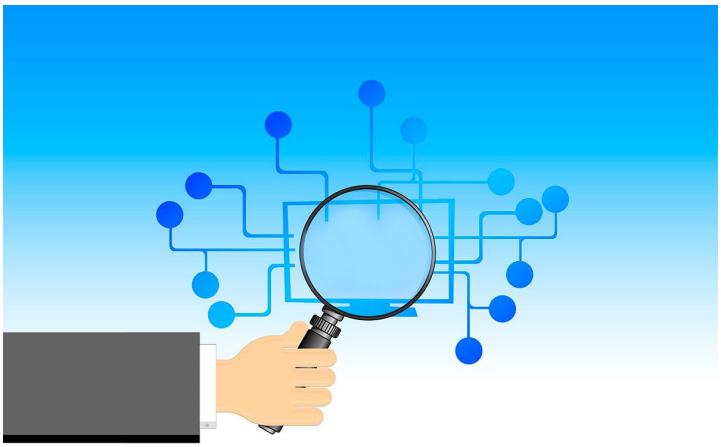
No mapeamento lógico-conceitual, cada entidade vira tabela. O mesmo ocorrerá no LA, por ter cardinalidade máxima N:N. O relacionamento EL, por ter cardinalidade máxima 1:N será implementado na tabela LIVRO.

MÓDULO 4

- Identificar os aspectos físicos para implementação do modelo no SGBD

CONSULTAS

A partir de agora, conheceremos diretrizes que devem ser consideradas quando formos implementar um banco de dados relacional. As diretrizes abrangem aspectos que influenciam no desempenho do banco de dados. Assim, o projeto lógico pode sofrer ajustes para adaptar-se ao sistema gerenciador de banco de dados (SGBD) escolhido para a implementação.



Fonte: geralt/Pixabay

Planejar um banco de dados que tenha um bom desempenho pressupõe adquirir conhecimento sobre as consultas e transações que serão realizadas pela aplicação.

Um SGBD tipicamente processa e devolve dados requisitados em consultas para diversas finalidades, tais como recuperação, inclusão, exclusão ou mesmo atualização de dados. As consultas são implementadas com o auxílio da linguagem SQL (do Inglês, *Structured Query Language* – linguagem de consulta estruturada).

DICA

Um código típico de consulta para recuperar dados em SQL envolve o comando SELECT com as cláusulas *FROM* e *WHERE* .

Por exemplo, dada a tabela DOCENTE (CODIGODOCENTE, NOME, SEXO), o código a seguir recupera os registros de todas as professoras:

```
SELECT CODIGODOCENTE, NOME
FROM DOCENTE
WHERE SEXO='F'
```

A primeira linha do comando informa ao SGBD as colunas que devem ser exibidas após o processamento da consulta. Na segunda, especificamos o nome da tabela que contém os dados. Finalmente, na terceira linha, adicionamos uma condição de filtro que será processada pelo SGBD para recuperar as linhas de interesse.

TRANSAÇÕES

Diversos SGBDs modernos permitem a especificação de operações de transação.

Uma transação corresponde a uma série de operações que, quando submetidas ao SGBD, devem ser consideradas como uma unidade lógica de trabalho. Isso significa que todas as operações que compõem uma transação precisam ser executadas. Caso contrário, é necessário serem canceladas e nenhuma modificação ocorrerá no banco de dados.

Por exemplo, em um processo de inscrição em disciplinas, em geral, o aluno tem a liberdade para compor seu quadro de horário de disciplinas, para, em seguida, confirmar inscrição em diversas matérias. Assim, a inscrição em disciplinas deve ser considerada como um único processo ou transação. Trata-se de um procedimento atômico: ou todas as operações são confirmadas ou nenhuma delas é realizada.

INDEXAÇÃO EM BANCO DE DADOS

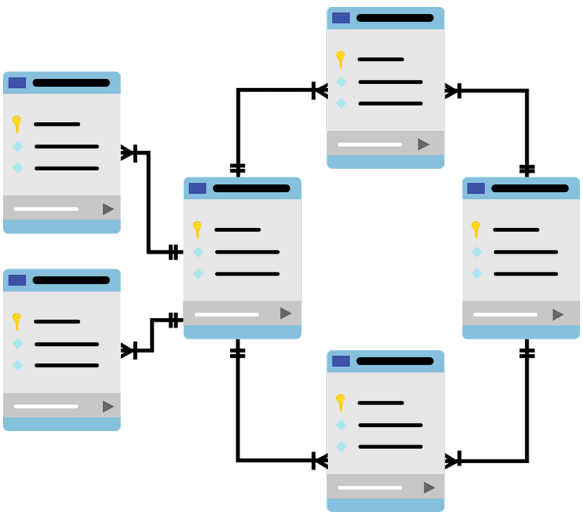
O desempenho de consultas é um assunto vasto que faz uso de diversas estratégias de acesso a dados, semelhantes às utilizadas em nosso dia a dia.

Por exemplo, ao buscarmos por determinada informação em algum livro, para *ganharmos tempo* , é comum primeiro consultar o índice remissivo do livro, que indicará a página onde se localiza o termo buscado.

📢 ATENÇÃO

Em banco de dados, índices funcionam como estruturas auxiliares utilizadas para tornar mais eficiente a recuperação de registros em resposta a determinadas condições de busca.

Normalmente, ao projetamos uma tabela com chave primária, os registros de dados são gravados em disco sem nenhum critério de ordenação das linhas da tabela. Para facilitar a consulta pelo valor da chave primária, o SGBD cria uma estrutura de índice para a chave primária de cada tabela.



Fonte: mcmurryjulie/Pixabay

A estrutura de índice poderá ser utilizada pelo SGBD caso seja necessário realizar consulta que envolva, por exemplo, uma condição de igualdade na coluna de chave primária da tabela. Quando isso ocorre, o desempenho da consulta em geral é melhor do que caso não existisse a estrutura de índice.

EXEMPLO PRÁTICO ENVOLVENDO INDEXAÇÃO:

Visando ressaltar a importância dos índices, realizamos um pequeno experimento, que consiste em submeter duas consultas ao SGBD, uma sem índice, e a segunda com uma coluna indexada.

Vamos perceber que, quando o SGBD processa uma consulta com o auxílio de um índice, o tempo de resposta tende a ser mais otimizado se comparado à execução da mesma consulta sem esse recurso.

Suponha então a existência de uma tabela DM_DOCENTE (CO_IES, NO_IES, CO_DOCENTE_IES, CO_MUNICIPIO_NASCIMENTO) - apresentada aqui com quatro colunas para fins de exemplo – originalmente, extraída do Censo da Educação Superior Brasileira de 2016.

A tabela contém 367.980 registros. Cada registro corresponde a um docente vinculado a uma instituição de ensino superior (IES). Ainda, originalmente, os registros de DM_DOCENTE estão fisicamente ordenados pela coluna CO_IES e a tabela não possui chave primária definida.

Nosso objetivo é recuperar todas as colunas da tabela, referentes ao docente que possui o valor 850516 para a coluna CO_DOCENTE_IES.

O comando SQL executado na consulta I a seguir, serve para esse propósito:

```
SELECT *
FROM DM_DOCENTE
WHERE CO_DOCENTE_IES=850516;
```

Essa consulta demorou **2,5** segundos para ser executada.

Agora, criaremos uma tabela chamada DM_DOCENTE_2, contendo os mesmos registros de DM_DOCENTE, no entanto com os registros ordenados pela coluna CO_DOCENTE_IES, conforme código SQL a seguir:

```
/*
Tabela DM_DOCENTE_2 com registros ordenados por
CO_DOCENTE_IES;
*/
CREATE TABLE DM_DOCENTE_2 AS
SELECT *
FROM DM_DOCENTE
ORDER BY CO_DOCENTE_IES;
```

Adicionaremos chave primária à tabela DM_DOCENTE_2, escolhendo a coluna CO_DOCENTE_IES, conforme código SQL a seguir:

```
/*
Ao adicionar chave primária na tabela DM_DOCENTE_2, o SGBD
cria um índice para a coluna CO_DOCENTE_IES.
*/
ALTER TABLE DM_DOCENTE_2 ADD PRIMARY KEY (CO_DOCENTE_IES);
```

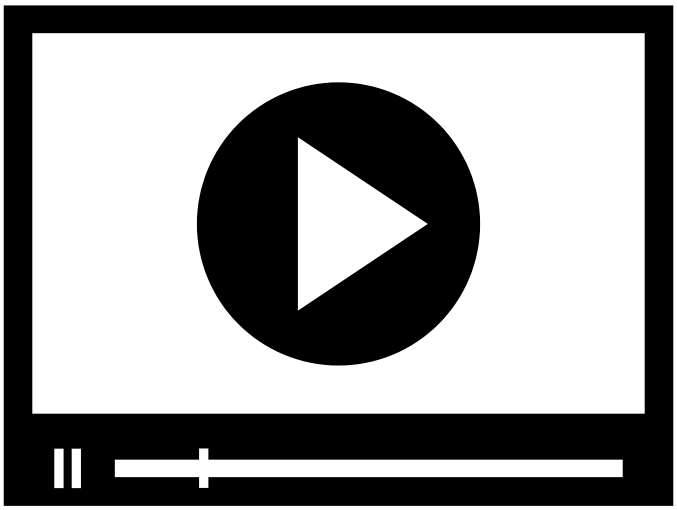
Finalmente, executaremos a consulta II:

```
SELECT *
FROM DM_DOCENTE_2
WHERE CO_DOCENTE_IES=850516;
```

Essa consulta demorou **0,06** segundos para ser executada.

O resultado do processamento das consultas é igual, uma vez que temos os mesmos registros em ambas as tabelas. Contudo, a consulta 2 foi processada com mais eficiência.

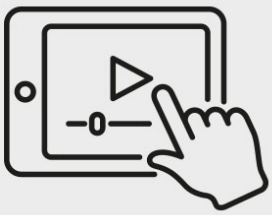
Após breve contextualização envolvendo consulta, transação e indexação, passaremos a estudar os fatores que influenciam no projeto de bancos de dados relacionais.



EXEMPLO PRÁTICO (INDEXAÇÃO)

Confira o vídeo mostrando a melhoria no desempenho de uma consulta em um banco de dados com indexação:

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



PROJETO FÍSICO EM BANCOS DE DADOS RELACIONAIS

Projetar um banco de dados é um processo que envolve as seguintes etapas:

Levantamento de requisitos

Projeto conceitual

Projeto lógico

Projeto físico

Projetar fisicamente um banco de dados é o mesmo que “colocar a mão na massa”, ou seja, acessar recursos do sistema gerenciador de banco de dados (SGBD) para atividades de criação da estrutura física do banco, que, na maioria das vezes, ocorre com o auxílio de alguma ferramenta CASE capaz de gerar códigos na linguagem SQL para criar as tabelas, relacionamentos e demais componentes do banco de dados.

É comum que haja mais de uma alternativa para implementar um banco de dados tomando como base o esquema conceitual. Ainda, o projeto físico de banco de dados é comumente influenciado pelos seguintes fatores:

CONSULTAS E TRANSAÇÕES DE BANCO DE DADOS

É necessário planejar as consultas e transações que deverão ocorrer no banco de dados. De um modo geral, para cada consulta de recuperação de dados, é necessário conhecer:

As tabelas acessadas pela consulta;

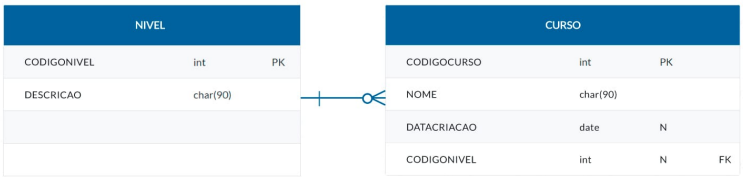
As colunas que serão utilizadas em condições de seleção;

A natureza da condição de seleção: intervalo, igualdade ou desigualdade;

Colunas utilizadas na composição de operações de junção;

Colunas cujos valores aparecerão nos resultados da consulta.

Em se tratando de tabelas de um banco de dados, é comum criarmos índices para determinadas colunas. Em especial, colunas relacionadas aos itens 2 e 4 são boas candidatas para serem indexadas.



Fonte: Autor

Figura: Tabelas NIVEL e CURSO.

SELECT *
FROM CURSO
WHERE NOME='Medicina' **OR** NOME='Nutrição';

O SGBD precisa avaliar se há algum registro na tabela CURSO cujo conteúdo da coluna NOME seja “Medicina” ou “Nutrição.” Trata-se de uma consulta enquadrada no item 2: há uma condição de seleção na cláusula WHERE envolvendo a coluna NOME: uma boa candidata para criação de índice.

Veja a consulta II a seguir, que objetiva recuperar o nome do curso e o nível ao qual ele pertence:

SELECT NOME, DESCRICAO
FROM CURSO **JOIN** NIVEL **ON** (CURSO.CODIGONIVEL=NIVEL.CODIGONIVEL);

A consulta usa um comando de junção (JOIN). A condição (CURSO.CODIGONIVEL=NIVEL.CODIGONIVEL) será avaliada diversas vezes ao longo do processamento da consulta. Trata-se de uma consulta enquadrada no item 4: as colunas CODIGONIVEL presentes na tabela são boas candidatas para criação de índices.

No caso de operações de atualização de dados, é necessário conhecer:

As tabelas alvo da atualização;

A categoria da atualização em cada tabela: exclusão, atualização ou inserção;

Colunas utilizadas em condições de seleção para exclusão ou atualização;

Colunas alvo das operações de atualização.

Ainda no contexto de indexação, colunas relacionadas ao item 3 são boas candidatas para serem indexadas. Ao mesmo tempo, o ideal é não criar índices para as colunas relacionadas ao item 4. Vamos estudar um exemplo?

Veja a consulta III a seguir, que objetiva excluir todos os cursos que tenham a *string* “Engenharia”.

DELETE FROM CURSO
WHERE CURSO LIKE '%ENGENHARIA%';

O SGBD precisa localizar os registros para então apagá-los do banco de dados. Para tanto, executará a condição de seleção presente no WHERE. Trata-se de uma consulta enquadrada no item 3: a coluna NOME presente na tabela é boa candidata para criação de índice.

FREQUÊNCIA DE CHAMADA DE CONSULTAS E TRANSAÇÕES ESPERADA

Vimos a importância de identificar detalhes sobre as consultas de recuperação e transações de atualização esperadas. No entanto, saber a respeito da frequência de uso esperada para operações de consulta e transações também é uma boa estratégia para obter desempenho.

📢 ATENÇÃO

Aplicando-se a “regra do 80/20”, conhecida como **Princípio de Pareto**, em um sistema de banco de dados, estima-se que 80% do processamento é originado de somente 20% das consultas e transações. Por isso, é rara a necessidade de coletar informações estatísticas completas e taxas de chamada para todas as consultas e transações, bastando priorizar 20% das mais relevantes.

PRINCÍPIO DE PARETO

O princípio de Pareto (também conhecido como regra do 80/20) afirma que, para muitos eventos, aproximadamente 80% dos efeitos vêm de 20% das causas.

RESTRIÇÕES DE TEMPO DE CONSULTA E TRANSAÇÕES

Dependendo da natureza da aplicação, podem existir consultas e transações com restrições de desempenho bastante rigorosas. Para exemplificar, poderia existir a restrição de que uma transação de compras tenha que terminar o seu processamento de pagamento dentro de sete segundos em 90% das vezes em que é chamada, e que ela nunca deve ultrapassar quinze segundos.

Essas restrições referentes ao tempo têm forte impacto nas colunas candidatas a serem indexadas. Em especial, tais colunas devem ser priorizadas quando da decisão da criação de índices para as tabelas.

FREQUÊNCIAS ESPERADAS DE OPERAÇÕES DE ATUALIZAÇÃO

Se a tabela é atualizada com frequência, deve-se evitar a criação de índices nas colunas, pois a atualização de colunas indexadas, frequentemente, requer atualização na estrutura de índice.

★ EXEMPLO

Por exemplo, se uma tabela possui cinco colunas indexadas, a inserção de um novo registro requer a atualização dos índices, o que pode causar lentidão nesse tipo de operação.

RESTRIÇÕES DE EXCLUSIVIDADE EM COLUNAS DA TABELA

É útil criar índice para cada coluna com restrição de unicidade na tabela. Em uma operação típica de inserção, o SGBD pode validar essa restrição de exclusividade fazendo consulta à estrutura de índice, rejeitando a inserção caso o valor da coluna seja encontrado no índice.

CONSULTAS ENVOLVENDO MAIS DE UMA TABELA

Você perceberá que, em geral, a maior parte das consultas para recuperar informações de um banco de dados envolve diversas tabelas. Isso ocorre, principalmente, quando o projeto leva em conta as regras de normalização.

Considere a estrutura de duas tabelas relacionadas, conforme a seguir:

MUNICIPIO (CO_MUNICIPIO, NOME)
DM_DOCENTE_2 (CO_DOCENTE_IES, CO_IES, NO_IES, CO_MUNICIPIO_NASCIMENTO)
CO_MUNICIPIO_NASCIMENTO REFERENCIA MUNICIPIO

A relação entre as tabelas está representada pela coluna de chave estrangeira CO_MUNICIPIO_NASCIMENTO da tabela DM_DOCENTE_2, a qual faz referência para a coluna chave primária CO_MUNICIPIO da tabela MUNICIPIO.

Nosso objetivo é recuperar o código do docente e nome do município de nascimento dele.

Perceba que as colunas alvo do resultado estão presentes em tabelas distintas: NOME, na tabela MUNICIPIO e CO_DOCENTE_IES, na tabela DM_DOCENTE_2.

O código em SQL que recupera os dados de interesse está expresso a seguir:

SELECT CODIGO_DOCENTE_IES, NOME
FROM DM_DOCENTE_2, MUNICIPIO
WHERE
MUNICIPIO.CO_MUNICIPIO=DM_DOCENTE_2.CO_MUNICIPIO_NASCIMENTO;

A primeira linha do comando serve para declararmos as colunas que farão parte do resultado da consulta. Na segunda, informamos as tabelas de interesse. Finalmente, na última linha, há uma condição de filtro, envolvendo uma igualdade entre a chave primária da tabela MUNICIPIO e a chave estrangeira da tabela DM_DOCENTE_2.

Para processar a consulta anterior, o SGBD cria uma estrutura de tabela temporária que contém a combinação de cada linha da tabela DM_DOCENTE_2 com cada linha da tabela MUNICIPIO. Se considerarmos os 367.980 registros de DM_DOCENTE_2 e os 5.570 registros da tabela MUNICIPIO, a tabela temporária teria mais de dois bilhões de linhas (367.980*5.570). Finalmente, a partir da tabela temporária, o SGBD executa o filtro especificado no comando WHERE para então exibir as colunas listadas no comando SELECT. O processo anterior é bastante custoso para o SGBD, ainda que cada sistema internamente use técnicas para otimizar o processamento.

📢 ATENÇÃO

Note que, se esse tipo de consulta for frequente, haverá grande probabilidade de lentidão no sistema.

A seguir, apresentaremos uma alternativa para minimizar esse custo, no entanto tendo como consequência a introdução de algum nível de redundância nos dados.

DESNORMALIZAR PARA GANHAR DESEMPENHO

Quando um esquema de banco de dados está normalizado até a 3FN, os problemas com redundância de dados são minimizados, pois, em geral, existe uma tabela para cada objeto modelado.

Fonte: John Schnobrich/Unsplash class=

Ao mesmo tempo, vimos que processar a consulta anterior requer acesso às duas tabelas para recuperar as informações - o que gera um custo adicional de processamento.

Logo, se quisermos priorizar desempenho, teremos que sacrificar as vantagens de um modelo normalizado. Esse processo é conhecido por desnormalização.

Nossa intenção a partir de agora é gerar uma estrutura que permita obter os mesmos resultados da consulta anterior, no entanto, usando somente uma tabela. Esse tipo de situação é comum quando temos a necessidade de produzir relatórios em um sistema.

Ao desnormalizar o modelo, ficamos com a seguinte tabela: DM_DOCENTE_2 (CO_DOCENTE_IES, CO_IES, NO_IES, CO_MUNICIPIO_NASCIMENTO, NOME). Note que a coluna NOME é dependente da coluna CO_MUNICIPIO_NASCIMENTO, ou seja, precisamos ter em mente que estamos diante de uma dependência funcional parcial, violando a 2FN.

Diante da nova estrutura, o código a seguir recupera as informações, agora envolvendo somente uma tabela:

```
SELECT CODIGO_DOCENTE_IES, NOME
FROM DM_DOCENTE_2;
```

📢 ATENÇÃO

O processo de desnormalização deve ser planejado com critério, visto que, ao mesmo tempo em que há potencial de ganho em relação ao desempenho de determinadas consultas, a desnormalização introduz redundância nos dados e, ao mesmo tempo, é necessária atualização adicional visando manter a consistência das colunas redundantes.

Ao longo deste módulo, percebemos que a criação do modelo físico em um SGBD está atrelada ao objetivo de criar um banco de dados de maneira que problemas de baixo desempenho sejam evitados. Para isso, é necessário mapear as principais consultas e transações a serem processadas ao longo do ciclo de vida do banco de dados.

VERIFICANDO O APRENDIZADO

1. ACERCA DO PROCESSO DE DESNORMALIZAÇÃO, ASSINALE A PROPOSIÇÃO VERDADEIRA.

A) Ao desnormalizar um modelo da 3FN para a 2FN, introduz-se redundância de dados, tendo vantagem a diminuição do desempenho no acesso aos dados.

B) A desnormalização nunca introduz redundância de dados.

C) Ao desnormalizar um modelo da 3FN para a 2FN, introduz-se redundância de dados, tendo vantagem aumento do desempenho no acesso aos dados.

D) Ao desnormalizar um esquema de banco de dados, o número de tabelas do modelo é triplicado.

2. OS ÍNDICES SÃO USADOS EM BANCOS DE DADOS COM A MESMA FINALIDADE DOS ÍNDICES DE LIVROS, ISTO É, PARA ACELERAR A CONSULTA A DETERMINADOS DADOS. COMO SE TRATAM DE ESTRUTURAS AUXILIARES QUE OCUPAM ESPAÇO E GERAM REDUNDÂNCIA DE DADOS, É PRECISO CRITÉRIO NA ESCOLHA DOS DADOS A SEREM INDEXADOS.
QUAL DAS ALTERNATIVAS ABAIXO MENCIONA ELEMENTOS DE UM BANCO DE DADOS QUE SÃO CANDIDATOS A SEREM INDEXADOS?

A) Colunas mais usadas em condições de seleção.

B) Tabelas mais acessadas por consultas.

C) Linhas mais selecionadas em consultas.

D) Células que aparecem mais vezes em resultados de consulta.

GABARITO

1. Acerca do processo de desnormalização, assinale a proposição verdadeira.

A alternativa **"C "** está correta.

De fato, quando aplicamos a desnormalização em um banco de dados, o conteúdo de uma ou mais tabelas é movido para outra tabela. Com isso, certa redundância é inserida no sistema. Por outro lado, consultas que, anteriormente, envolviam diversas tabelas, em geral passam a ser executadas diretamente na tabela desnormalizada

2. Os índices são usados em bancos de dados com a mesma finalidade dos índices de livros, isto é, para acelerar a consulta a determinados dados. Como se tratam de estruturas auxiliares que ocupam espaço e geram redundância de dados, é preciso critério na escolha dos dados a serem indexados.
Qual das alternativas abaixo menciona elementos de um banco de dados que são candidatos a serem indexados?

A alternativa **"A "** está correta.

Para acelerar o processamento de consultas ao banco de dados, uma das técnicas mais usadas é criar índices para as colunas de tabelas que mais aparecem nas cláusulas WHERE dos comandos de consulta SELECT.

CONCLUSÃO

CONSIDERAÇÕES FINAIS

Este tema iniciou com o estudo dos componentes do modelo relacional de banco de dados. Além das tabelas e seus componentes, foram apresentadas as definições de chave primária e de chave estrangeira como elementos importantes na implementação de relacionamentos.

Investigamos as três primeiras formas normais, como modos de construir modelos livres das redundâncias que aparecem em algumas dependências funcionais. Depois, analisamos uma série de regras usadas para o mapeamento do modelo conceitual para o modelo lógico.

Finalmente, investigamos diretrizes que devem ser levadas em consideração quando da implementação do modelo físico em um SGBD.

Para ouvir um *podcast* sobre o assunto, acesse a versão online deste conteúdo.

REFERÊNCIAS

ELMASRI, R.; NAVATHE, S. **Sistemas de Banco de Dados**. 7.Ed. São Paulo: Pearson, 2019.

HEUSER, Carlos A. **Projeto de Banco de Dados**. 6.Ed. Porto Alegre: Bookman, 2009.

SOUZA, Odécio. **Edgar Frank Codd and the Relational Database: a contribution to the History of Computing**. 2015. 155 f. Dissertação (Mestrado em História da Ciência) - Pontifícia Universidade Católica de São Paulo, São Paulo, 2015.

EXPLORE+

Para aprofundar seu conhecimento sobre a ferramenta BrModelo, acesse o portal GitHub e obtenha informações sobre a correção de bugs e outras funcionalidades.

Caso tenha interesse em continuar estudando as ferramentas comerciais, acesse o site Vertabelo. Observe como funcionam as ferramentas de modelagem, as quais permitem o uso de alguma notação gráfica para representar um banco de dados relacional, conforme mencionado na seção **Esquema diagramático de banco de dados relacional**.

Para complementar seu estudo, leia o texto *Edgar Frank Codd e o Banco de Dados Relacional: uma contribuição para a História da Computação* , de Odécio Souza, publicado pela Pontifícia Universidade Católica de São Paulo, São Paulo, 2015. O material fala sobre o modelo relacional que foi introduzido por Ted Codd, pesquisador visionário que apresentou em 1970 as bases científicas sobre as quais a maior parte dos SGBDs relacionais fazem uso. Trata-se de um trabalho completo sobe as contribuições de Codd para a Ciência da Computação.

Recomendamos ainda que você leia na referência (ELMASRI e NAVATHE, 2019) o capítulo que trata sobre indexação. O desempenho de um banco de dados relacional em termos de recuperação de informações a partir de determinada consulta tem relação direta com o projeto dos mecanismos de indexação.

CONTEUDISTA

Nathielly de Souza Campos

🔗 CURRÍCULO LATTES